

Generalized Sequential Gaussian Simulation on Group Size v and Screen-Effect Approximations for Large Field Simulations¹

Roussos Dimitrakopoulos² and Xiaochun Luo³

The modelling of spatial uncertainty in attributes of geological phenomena is frequently based on the stochastic simulation of Gaussian random fields. This paper presents a generalization of the sequential Gaussian simulation method founded upon the group decomposition of the posterior probability density function of a stationary and ergodic Gaussian random field into posterior probability densities of a set of groups of nodes of size v . The method, which is termed "generalized sequential Gaussian simulation on group size v ," relies computationally on sharing the neighborhood of adjacent nodes and simulates groups of v nodes at a time, instead of the traditional node-by-node simulation. The new method is computationally efficient and suitable for simulation on large grids of nodes. Results suggest that, in terms of computing cost (time), an optimal size v of a group is about 80% of the optimal neighborhood used for sequential Gaussian simulation and that computation can be up to 50 times faster than the regular sequential Gaussian method, with little loss in accuracy. The effectiveness of the method is assessed by using a general measure of accuracy, "screen-effect approximation loss" (SEA loss), defined herein as the mean-square difference between the simulated value posterior to the information in the neighborhood and the simulated value posterior to all information, and shown to be determined by the corresponding posterior variances. The results presented show that both the exponential and the spherical models perform well and can meet the less-than 5% relative SEA loss requirement for any grid setup using a relatively small neighborhood. The Gaussian covariance model was found to have a relatively high relative SEA loss in most cases.

KEY WORDS: conditional stochastic simulation, large field simulation, uncertainty modelling.

INTRODUCTION

The modelling and assessment of spatial uncertainty in the attributes of geological phenomena are frequently based on the stochastic simulation of Gaussian random fields conditional to available data. Sequential conditional simulation (Alabert, 1987; Journel, 1994) is a well-known approach based on the decomposition of

¹Received 20 September 2002; accepted 4 February 2004.

²WH Bryan Mining Geology Research Centre, The University of Queensland, Brisbane Qld 4072, Australia; e-mail: roussos@uq.edu.au

³Department of Mining and Minerals Engineering, McGill University, Montreal, Quebec, Canada.

the multivariate probability density function of a stationary and ergodic random process, $Z(\mathbf{u})$, $\mathbf{u} \in R^n$, to the product of univariate posterior distribution functions (Johnson, 1987; Ripley, 1987; Rossenblatt, 1952; Rubinstein, 1981; Scheuer and Stoller, 1962). In the simplest case, sequential simulation is as follows. Let $f(z_1, z_2)$ be a probability distribution function associated with a bivariate process $Z = \{Z_1, Z_2\}$ at $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2\}$. Generating realization $\mathbf{z} = \{z_1, z_2\}$ is then based on the product decomposition of univariate posterior probability density functions $f(z_1, z_2) = f(z_1)f(z_2|z_1)$, or Bayes relation, where $f(z_2|z_1)$ is the posterior distribution of Z_2 , given $Z_1 = z_1$. To generate a realization $\mathbf{z} = \{z_1, z_2\}$ of $Z = \{Z_1, Z_2\}$, a value z_1 is drawn for Z_1 based on $f(z_1)$; then z_2 is drawn from the posterior probability function $f(z_2|z_1)$. This sequential principle remains the same for the N -variate process, as discussed in a subsequent section. When $Z(\mathbf{u})$ is Gaussian, the resulting method is termed “sequential Gaussian simulation” or SGS (Isaaks, 1990; Journel, 1994). SGS employs simple kriging at a node to estimate the posterior mean and variance, with random sampling of the posterior distribution to generate a realization at the corresponding node. SGS provides a relatively simple, efficient, and widely used conditional simulation algorithm. It is useful in generating relatively large simulations in the industrial environment, where computational efficiency and effective implementation are important (e.g., Ravenscroft, 1994). In this context, large simulations are considered to be those on grids with up to 10^8 nodes (e.g., Omre, Solna, and Tjelmeland, 1993).

The computational efficiency of a conditional simulation algorithm becomes important, for example, when simulating spatial attributes of geological phenomena represented by grids in the order of hundreds of millions of nodes and assessing risk through multiple realizations (e.g., Godoy, 2003). To assess “computational costs” for various conditional simulation approaches and their suitability for large problems, the number of floating point operations (flops) required for computations can serve as a measure for comparison. Consider, for example, the simulation of values at each of the N nodes of a grid. The number of floating point operations, or flops, required over the grid is a function of N , denoted herein as $O(N)$, and reads “in the order of N .” If \mathbf{C} is a matrix associated with a 3D grid of $N = 10^6$, and the linear system $\mathbf{C}\mathbf{w} = \mathbf{D}$, where \mathbf{w} and \mathbf{D} are vectors, needs to be solved, the computational cost in the absence of any structure in \mathbf{C} is $O(N^3) = O(10^{18})$ flops, that is, a million, million, million flops (Cormen, Leiserson, and Rivest, 1990). With today’s teraflop supercomputing machines this would take days to compute. If the computational cost was, say, $O(N \log N) = O(6 \times 10^6)$ flops, computations would be executed in a fraction of a second on any megaflop PC.

Issues related to stochastic simulation methods for large problems are discussed in the technical literature, including Omre, Solna, and Tjelmeland (1993), who present the sequential simulation of Gaussian and Gaussian intrinsic random functions based on Markov properties of random fields. The approach leads to Markov properties justifying the selection of few “local” data for conditioning,

and otherwise it is similar to SGS. Dietrich and Newsam (1996) show that the stationary random process covariance matrix \mathbf{C} can be embedded into a larger block circulant matrix \mathbf{M} with square root \mathbf{S} (i.e., $\mathbf{M} = \mathbf{S}^T\mathbf{S}$) and realization $\mathbf{S}\mathbf{e}$, where \mathbf{e} is white noise. Their circulant embedding approach uses fast Fourier transforms for \mathbf{S} and generates exact realizations at a cost of $O(N \log N)$ flops. The computational improvement over, for example, the well-known LU decomposition $\mathbf{C} = \mathbf{L}^T\mathbf{L}$ of the covariance matrix \mathbf{C} (Davis, 1987a,b) is substantial. LU requires $O(N^3)$ flops and may only be marginally improved (Dowd and Saraç, 1994).

In practice, sequential Gaussian simulation, like most simulation algorithms used in earth science and engineering problem solving, uses local neighborhoods to estimate the local posterior probability density functions. This neighborhood implementation substantially reduces computing cost and storage requirements and is based on screening effects the closest samples have over the data at larger distances, similarly to the screening effects in kriging (David, 1977). As discussed in the next section, the computational costs for SGS are reduced from $O(N^4)$ when all data is used to $O(N\nu_{\max}^3)$, where ν_{\max} is the size of the neighborhood. This is a substantial reduction in computational costs (Luo, 1998) and it provides one of the justifications for the wide use of the method.

Against this background, this paper suggests that computational efficiency can be further enhanced. This improvement starts with the observation that, in the implementation of SGS in a node-by-node sequential process, there is overlapping of neighborhoods amongst closest nodes, as shown in Figure 1. It is rational to consider sharing neighborhoods amongst adjacent grid nodes. This consideration leads to the decomposition of the multivariate probability density function of $Z(\mathbf{u})$, into groups of products of univariate posterior distributions, where each group is

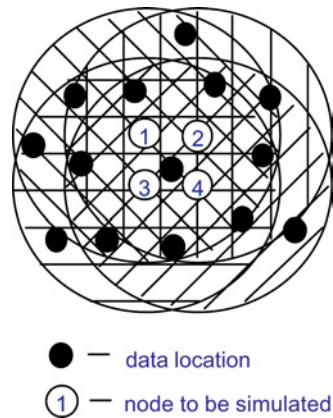


Figure 1. Four overlapping neighborhoods of a group of four nodes.

used to simultaneously generate realizations at the corresponding grid nodes. The group decomposition of the multivariate probability density function is a generalization of the sequential Gaussian simulation method, in which a decomposition with a single node in a group is identical to SGS, and a decomposition with all nodes in one group is identical to the LU method.

The screen-effect approximation (SEA) is used in all the above implementations and the trade-off between computational efficiency and accuracy needs to be quantified. This paper introduces a measure of the loss from SEA, or screen-effect approximation loss (SEA loss), defined as a mean-square difference between the simulated value posterior to the information in the neighborhood and the simulated value posterior to all information, and found to be a function of the corresponding posterior variances. SEA loss can be used to assess the optimal size of the neighborhood for different requirements.

In the first of the following sections, the pertinent aspects of the sequential Gaussian simulation are revisited. The proposed generalization of SGS is then detailed, including the main computational aspects. In the following section, *SEA loss* is defined, its properties are described, and loss for various covariance models and grid characteristics is assessed. The optimal size neighborhood for various covariance models and a brief example of the proposed generalized SGS are then presented. The final section presents the conclusions.

SEQUENTIAL GAUSSIAN SIMULATION REVISITED

Following the geostatistical terminology, consider the stationary Gaussian random field $Z(\mathbf{u}_i)$, $\mathbf{u}_i \in \mathbb{R}^n$, $i = 1, \dots, N$, also denoted as Z_i , indexed on a discrete grid D_N , and a set of conditioning data $\mathbf{d}_n = \{z(\mathbf{u}_\alpha), \alpha = 1, \dots, n\}$. In addition, consider for convenience the set Λ_i such that, $\Lambda_0 = \{\mathbf{d}_n\}$, $\Lambda_i = \Lambda_{i-1} \cup \{Z_i\}$; for example, $\Lambda_1 = \{\mathbf{d}_n, Z_1\}$, $\Lambda_2 = \{\mathbf{d}_n, Z_1, Z_2\}$, $\Lambda_3 = \{\mathbf{d}_n, Z_1, Z_2, Z_3\}$. Following the above notation, the sequential Gaussian simulation on D_N is based on the sampling from the N -variate distributions posterior to the data set Λ_0 (Journal, 1994),

$$F(\mathbf{u}_1, \dots, \mathbf{u}_N; z_1, \dots, z_N \mid \Lambda_0) = P(Z(\mathbf{u}_1) \leq z_1, \dots, Z(\mathbf{u}_N) \leq z_N \mid \Lambda_0) \quad (1)$$

with a density function equal to the product of the N single-variate posterior probability density functions

$$\begin{aligned} f(\mathbf{u}_1, \dots, \mathbf{u}_N; z_1, \dots, z_N \mid \Lambda_0) &= f(\mathbf{u}_N; z_N \mid \Lambda_{N-1}) \dots f(\mathbf{u}_1; z_1 \mid \Lambda_0) \\ &= \prod_{i=1}^N f(\mathbf{u}_i; z_i \mid \Lambda_{i-1}) \end{aligned} \quad (2)$$

The posterior probability density functions in Equation (2) can be seen as an extension of Bayes relation discussed in the previous section and are given by

$$f(\mathbf{u}_i; z_i | \Lambda_{i-1}) = N(E\{Z(\mathbf{u}_i) | \Lambda_{i-1}\}, \text{Var}\{Z(\mathbf{u}_i) | \Lambda_{i-1}\}) \quad (3)$$

where $N(\cdot)$ denotes the Gaussian probability density function of $Z(\mathbf{u}_i)$ with mean $E\{Z(\mathbf{u}_i) | \Lambda_{i-1}\}$, and variance $\text{Var}\{Z(\mathbf{u}_i) | \Lambda_{i-1}\}$ conditional to data set Λ_{i-1} . Realizations of Z_i are generated from the operation

$$Z(\mathbf{u}_i | \Lambda_{i-1}) = E\{Z(\mathbf{u}_i) | \Lambda_{i-1}\} + \sqrt{\text{Var}\{Z(\mathbf{u}_i) | \Lambda_{i-1}\}} \cdot \mathbf{e}_i \quad (4)$$

where all \mathbf{e}_i are independent, $N(0, 1)$ distributed random numbers; and the posterior mean and variance are given by

$$E\{Z(\mathbf{u}_i) | \Lambda_{i-1}\} = m_i + \mathbf{C}_{i\Lambda_{i-1}} \mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1} (\mathbf{Z}_{\Lambda_{i-1}} - \mathbf{m}_{\Lambda_{i-1}}) \quad (5)$$

and

$$\text{Var}\{Z(\mathbf{u}_i) | \Lambda_{i-1}\} = C_{ii} - \mathbf{C}_{i\Lambda_{i-1}} \mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1} \mathbf{C}_{\Lambda_{i-1}i} \quad (6)$$

where m_i and C_{ii} are the prior mean and variance of $Z(\mathbf{u}_i)$, $\mathbf{m}_{\Lambda_{i-1}}$ and $\mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1}$ are the prior means and covariances of the data, respectively; vector $\mathbf{Z}_{\Lambda_{i-1}}$ denotes the data, and $\mathbf{C}_{i\Lambda_{i-1}}$ is the covariance between $Z(\mathbf{u}_i)$ and data set Λ_{i-1} .

The implementation of sequential Gaussian simulation is as follows:

1. define a random path visiting the N nodes of grid D_N to be simulated;
2. at a node \mathbf{u}_i of D_N , generate a value using Equation (4);
3. add the simulated value into the data set; and
4. loop and repeat Steps 2 and 3 until all N nodes are simulated.

Computationally, SGS as described above needs $O((n+i-1)^3)$ floating point operations for $\mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1}$ (Steinberg, 1974), and the computing cost for Equations (5) and (6) is $O((n+i-1)^3 + (n+i-1)^2 + 2(n+i-1))$ flops. This is in practice $O((n+i-1)^3) = O((n+i)^3)$ and the overall computing cost for the N steps in SGS is $\sum_{i=1}^N O((n+1)^3) = O((n+N)^4)$.

The minimum memory allocation required (storage requirement) for the SGS algorithm is determined by the maximum size of the matrix $\mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1}$, in the order of $(n+N-1)^2$. This storage requirement deems the SGS algorithm as described above to be impractical since $n+N$ is usually in the order of hundreds of thousands to tens of millions of nodes. In practice, the implementation of SGS is based on the so-called “screen-effect approximation” discussed next.

SGS With a Screen-Effect Approximation

The implementation of SGS is based on fixed-size neighborhoods where the posterior probability density function at a node posterior to all data is approximated from the data within the neighborhood closest to this node. SEA on Equation (2) is then

$$f(\mathbf{u}_1, \dots, \mathbf{u}_N; z_1, \dots, z_N | \Lambda_0) \approx \prod_{i=1}^N f(\mathbf{u}_i; z_i | \lambda_{i-1}) \quad (7)$$

where λ_{i-1} is the data set in the neighborhood of \mathbf{u}_i and $\lambda_{i-1} \subseteq \Lambda_{i-1}$, and realizations of Z_i are generated from

$$Z(\mathbf{u}_i | \Lambda_{i-1}) \approx E\{Z(\mathbf{u}_i) | \lambda_{i-1}\} + \sqrt{\text{Var}\{Z(\mathbf{u}_i) | \lambda_{i-1}\}} \cdot \mathbf{e}_i \quad (8)$$

where all \mathbf{e}_i are independent, $N(0, 1)$ distributed random numbers; and the posterior mean and variance are now given by

$$E\{Z(\mathbf{u}_i) | \lambda_{i-1}\} = m_i + \mathbf{C}_{i\lambda_{i-1}} \mathbf{C}_{\lambda_{i-1}\lambda_{i-1}}^{-1} (\mathbf{Z}_{\lambda_{i-1}} - \mathbf{m}_{\lambda_{i-1}}) \quad (8a)$$

and

$$\text{Var}\{Z(\mathbf{u}_i) | \lambda_{i-1}\} = C_{ii} - \mathbf{C}_{i\lambda_{i-1}} \mathbf{C}_{\lambda_{i-1}\lambda_{i-1}}^{-1} \mathbf{C}_{\lambda_{i-1}i} \quad (8b)$$

where m_i and C_{ii} are the prior mean and variance of $Z(\mathbf{u}_i)$; $\mathbf{m}_{\lambda_{i-1}}$ and $\mathbf{C}_{\lambda_{i-1}\lambda_{i-1}}^{-1}$ are the prior means and inverse covariances of the data λ_{i-1} in the neighborhood of \mathbf{u}_i , respectively; vector $\mathbf{Z}_{\lambda_{i-1}}$ denotes the data and $\mathbf{C}_{i\lambda_{i-1}}$ is the covariance between $Z(\mathbf{u}_i)$ and data set λ_{i-1} .

Sequential Gaussian simulation with SEA is then based on Equation (8) instead of (4). By defining an upper bound of the neighborhood size, say v_{\max} , the computing cost is $O(Nv_{\max}^3)$ flops instead of $O((n+N)^4)$. Note that v_{\max} is usually in the order of 10–30 nodes, i.e., small compared to n , the total number of data. Similarly, the storage requirements are also drastically reduced compared to SGS based on (4).

A GENERALIZATION OF THE SEQUENTIAL GAUSSIAN SIMULATION

In practice, D_N is usually large and dense compared to the available data, indicating that the neighborhoods of closest nodes are overlapping as shown in

Figure 1. It is natural to consider generating realizations of Z_i simultaneously and in groups of close nodes, instead of the node-by-node sequential simulation process in SGS. This sequential simulation of groups of nodes takes advantage of the neighborhood sharing for the closest nodes and leads to computational efficiencies. By analogy to the current SGS method, which decomposes the posterior probability density into N posterior probability densities for the N nodes to be simulated, the method of decomposition in groups is a generalization of SGS and is as follows.

Partition D_{N_j} into k groups of $v_j, j = 1, \dots, k$, clustered nodes for each k group, and set $N_j = \sum_{i=1}^j v_i, j = 1, \dots, k, N = N_k$. Then, the decomposition of the posterior probability density in Equation (2) into posterior probability densities for k groups becomes

$$\begin{aligned}
 f(\mathbf{u}_1, \dots, \mathbf{u}_N; z_1, \dots, z_N \mid \Lambda_0) &= \prod_{i=1}^{N_1} f(\mathbf{u}_i; z_i \mid \Lambda_{i-1}) \cdots \prod_{i=N_{k-1}+1}^{N_k} f(\mathbf{u}_i; z_i \mid \Lambda_{i-1}) \\
 &= \prod_{j=1}^k \left\{ \prod_{i=N_{j-1}+1}^{N_j} f(\mathbf{u}_i; z_i \mid \Lambda_{i-1}) \right\} \tag{9}
 \end{aligned}$$

Considering SEA, similarly to the implementation of SGS in the previous section, we have

$$f(\mathbf{u}_1, \dots, \mathbf{u}_N; z_1, \dots, z_N \mid \Lambda_0) \approx \prod_{j=1}^k \left\{ \prod_{i=N_{j-1}+1}^{N_j} f(\mathbf{u}_i; z_i \mid \lambda_{i-1}) \right\} \tag{10}$$

The posterior mean vector and posterior covariance matrix of the j th group D_{N_j} ,

$$\prod_{i=N_{j-1}+1}^{N_j} f(\mathbf{u}_i; z_i \mid \lambda_{i-1}), \quad j = 1, \dots, k,$$

are

$$E\{\mathbf{Z}(\mathbf{u}_i^{N_j}) \mid \lambda_{i-1}\} = \mathbf{m}_j + \mathbf{C}_{j\lambda_{j-1}} \mathbf{C}_{\lambda_{j-1}\lambda_{j-1}}^{-1} (\mathbf{Z}_{\lambda_{j-1}} - \mathbf{m}_{\lambda_{j-1}}) \tag{11}$$

and

$$\text{Cov}\{\mathbf{Z}(\mathbf{u}_i^{N_j}) \mid \lambda_{i-1}\} = \mathbf{C}_{jj\lambda_{j-1}} = \mathbf{C}_{jj} - \mathbf{C}_{j\lambda_{j-1}} \mathbf{C}_{\lambda_{j-1}\lambda_{j-1}}^{-1} \mathbf{C}_{\lambda_{j-1}j} \tag{12}$$

where vector \mathbf{m}_j and $\mathbf{m}_{\lambda_{j-1}}$ denote the prior means of group $\mathbf{Z}(\mathbf{u}_i^{N_j})$ and of the set of data in λ_{i-1} , respectively; vector $\mathbf{Z}_{\lambda_{j-1}}$ denotes the data in the neighborhood; $\mathbf{C}_{\lambda_{j-1}\lambda_{j-1}}^{-1}$ denotes the inverse of the prior covariance matrix of λ_{i-1} ; \mathbf{C}_{jj} denotes the covariance matrix of $\mathbf{Z}(\mathbf{u}_i^{N_j})$; and $\mathbf{C}_{j\lambda_{j-1}}^T = \mathbf{C}_{j\lambda_{j-1}}$ denotes the prior covariances between $\mathbf{Z}(\mathbf{u}_i^{N_j})$ and λ_{i-1} .

Consider the Cholesky decomposition $\mathbf{C}_{jj\cdot\lambda_{j-1}} = \mathbf{L}\mathbf{L}^T$, where \mathbf{L} is the lower triangular matrix of $\mathbf{C}_{jj\cdot\lambda_{j-1}}$, then the simulated values of the j th group can be generated from the operation

$$\mathbf{Z}((\mathbf{u}_i^{N_j}) | \lambda_{i-1}) = \mathbf{m}_j + \mathbf{C}_{j\lambda_{j-1}} \mathbf{C}_{\lambda_{j-1}\lambda_{j-1}}^{-1} (\mathbf{Z}_{\lambda_{j-1}} - \mathbf{m}_{\lambda_{j-1}}) + \mathbf{L}\mathbf{e}_j \quad (13)$$

where the vector of independent random numbers $\mathbf{e}_j \sim N(\mathbf{0}, \mathbf{I})$.

If the number of nodes in each group is identical, say ν and $N = k\nu$, the above method can be seen as a generalization of SGS, termed “generalized SGS on group size ν ” or GSGS- ν . From Equation (13) it follows that when $\nu = 1$, GSGS-1 is identical to the sequential Gaussian simulation. When $\nu = N$ then, GSGS- N is identical to the LU decomposition method of Davis (1987a).

The implementation of the proposed generalized GSGS- ν algorithm on a group of size ν proceeds as follows:

1. define a path visiting each k group of the grid D_N and a path visiting the nodes in each group;
2. find a neighborhood for the current group;
3. calculate the posterior mean vector and the posterior covariance matrix of the current group from Equations (11) and (12);
4. generate the simulated values of the current group from Equation (13);
5. add the simulated values of the current group into the data set; and
6. loop until all groups are simulated.

An advantage of the proposed algorithm is its substantially reduced computing cost. Considering ν_{\max} as the maximum neighborhood size used in SGS, as in the previous section, the overall computing cost (time) for a discrete grid D_N of N nodes is

$$T_\nu(N) \approx O(k(\nu_{\max})^3 + k\nu^3) = O\left(\frac{N}{\nu}(\nu_{\max}^3 + \nu^3)\right) \quad (14)$$

Equation (14) shows that

- (i) the computing cost for D_N decreases as the size of the group ν is increasing from 1 to $\nu_{\max}/2^{1/3}$ or approximately 80% of ν_{\max} ;

- (ii) computing cost reaches a minimum when $\nu \approx 0.8\nu_{\max}$ and, using Equation (14), it gives

$$T_{0.8\nu_{\max}}(N) \approx 1.9N(\nu_{\max})^2 \approx \frac{1.9}{\nu_{\max}} T_1(N) \tag{15}$$

where $T_1(N) = O(N\nu_{\max}^3)$;

- (iii) computing cost is increasing when ν is increasing from 80% of ν_{\max} to N ;
- (iv) computing cost reaches the maximum when $\nu = N$ indicating $k = 1$ and $\nu_{\max} = \nu$,

$$T_N(N) \approx O(N^3)$$

To elucidate further, consider an example where the maximum size ν_{\max} is set to 100 and assume, without loss of generality, that the sizes of neighborhoods selected during the sequential process reach ν_{\max} in most cases, then the computing cost is

$$T_{0.8\nu_{\max}}(N) = T_{80}(N) \approx 1.9N100^2 \approx T_1(N)/50$$

From the above section, it can be seen that the computing cost for GSGS-1 is identical to that of SGS while the computing cost of GSGS- N is that of the LU decomposition. This is expected considering the equivalence of the methods. GSGS-1, or SGS, is relatively less efficient, and the computationally optimal approach is a GSGS- ν where ν is 80% of ν_{\max} . Figure 2 illustrates the effects of the group size ν and the neighborhood size ν_{\max} on the ratio of flops between

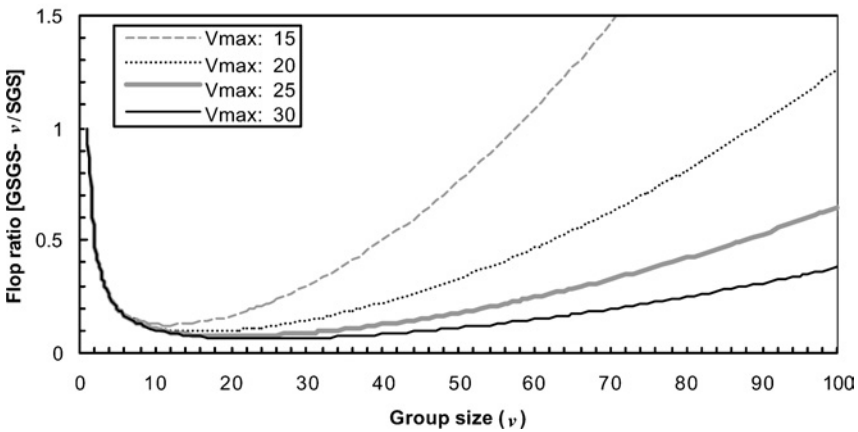


Figure 2. Ratio of flops between GSGS- ν and SGS versus the group size ν for various neighborhood sizes ν_{\max} .

GSGS- ν and SGS. The figure shows an abrupt decrease of flops for smaller group sizes due to computational gains, a relatively flat region around the minimum that suggests flexibility in selecting group sizes, and the slower rate of increase of computational costs as the group size increases after reaching a minimum.

THE SCREEN-EFFECT APPROXIMATION LOSS

Equation (15) shows that the computing cost of GSGS- ν mainly depends on the size of the neighborhood: the larger the size, the lower the computing cost, up to a given size. On the other hand, the larger size results in a larger difference between the simulated value posterior to the information in the neighborhood and the “ideal” value posterior to all information. This difference and the corresponding ability to balance precision requirements with computing cost raises the following questions: (i) how can the difference be measured; (ii) can the GSGS- ν match a given difference requirement; and (iii) what is the optimal size of the neighborhood under a given difference requirement?

This section addresses these questions, starting with the following definition of SEA loss.

The SEA loss at a node \mathbf{u}_i can be defined as the mean-square difference

$$\rho\{Z(\mathbf{u}_i) \mid \lambda_{i-1}, \Lambda_{i-1}\} = \frac{1}{2} E\{[Z(\mathbf{u}_i \mid \lambda_{i-1}) - Z(\mathbf{u}_i \mid \Lambda_{i-1})]^2\} \quad (16)$$

where $\rho\{\cdot\}$ denotes the SEA loss, $Z(\mathbf{u}_i \mid \lambda_{i-1})$ and $Z(\mathbf{u}_i \mid \Lambda_{i-1})$ denote $Z(\mathbf{u}_i)$ posterior to the data in the sets λ_{i-1} and Λ_{i-1} , respectively. Recall that $\lambda_{i-1} \subseteq \Lambda_{i-1}$.

From Equation (16) SEA loss is

$$\begin{aligned} \rho\{Z(\mathbf{u}_i) \mid \lambda_{i-1}, \Lambda_{i-1}\} &= \frac{1}{2} E\{[Z(\mathbf{u}_i \mid \lambda_{i-1}) - Z(\mathbf{u}_i \mid \Lambda_{i-1})]^2\} \\ &= \frac{1}{2} E\{[E\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} + \sqrt{\text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}} \cdot \mathbf{e}_i \\ &\quad - E\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\} - \sqrt{\text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}} \cdot \mathbf{e}_i]^2\} \\ &= \frac{1}{2} E\{[E\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} - E\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}]^2 \\ &\quad + \frac{1}{2} [\sqrt{\text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}} - \sqrt{\text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}}]^2\} \end{aligned}$$

Since

$$\begin{aligned} &E\{[E\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} - E\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}]^2\} \\ &= E\{[m_i + \mathbf{C}_{i\lambda_{i-1}} \mathbf{C}_{\lambda_{i-1}\lambda_{i-1}}^{-1} (\mathbf{Z}_{\lambda_{i-1}} - \mathbf{m}_{\lambda_{i-1}}) - m_i \\ &\quad - \mathbf{C}_{i\Lambda_{i-1}} \mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1} (\mathbf{Z}_{\Lambda_{i-1}} - \mathbf{m}_{\Lambda_{i-1}})]^2\} \end{aligned}$$

$$\begin{aligned}
 &= \mathbf{C}_{i\Lambda_{i-1}} \mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1} \mathbf{C}_{\Lambda_{i-1}i} + \mathbf{C}_{i\lambda_{i-1}} \mathbf{C}_{\lambda_{i-1}\lambda_{i-1}}^{-1} \mathbf{C}_{\lambda_{i-1}i} \\
 &\quad - 2\mathbf{C}_{\lambda_{i-1}\Lambda_{i-1}} \mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1} \mathbf{C}_{\Lambda_{i-1}\lambda_{i-1}} \mathbf{C}_{\lambda_{i-1}\lambda_{i-1}}^{-1} \mathbf{C}_{\lambda_{i-1}i} \\
 &= \mathbf{C}_{i\lambda_{i-1}} \mathbf{C}_{\lambda_{i-1}\lambda_{i-1}}^{-1} \mathbf{C}_{\lambda_{i-1}i} - \mathbf{C}_{i\Lambda_{i-1}} \mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1} \mathbf{C}_{\Lambda_{i-1}i} \\
 &\quad + 2(\mathbf{C}_{i\Lambda_{i-1}} \mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1} \mathbf{C}_{\Lambda_{i-1}i} - \mathbf{C}_{\lambda_{i-1}\Lambda_{i-1}} \mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1} \mathbf{C}_{\Lambda_{i-1}\lambda_{i-1}} \mathbf{C}_{\lambda_{i-1}\lambda_{i-1}}^{-1} \mathbf{C}_{\lambda_{i-1}i}) \\
 &= \text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} - \text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\} + 2(\mathbf{C}_{i\Lambda_{i-1}} \mathbf{C}_{\Lambda_{i-1}\Lambda_{i-1}}^{-1} \mathbf{C}_{\Lambda_{i-1}i \cdot \lambda_{i-1}})
 \end{aligned}$$

where $\mathbf{C}_{\Lambda_{i-1}i \cdot \lambda_{i-1}}$ denotes a vector of posterior covariances. Note that $\lambda_{i-1} \subseteq \Lambda_{i-1}$, and according to Property 1 of posterior covariances shown in the Appendix, $\mathbf{C}_{\Lambda_{i-1}i \cdot \lambda_{i-1}} = \mathbf{0}$. This leads to

$$E\{[E\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} - E\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}]^2\} = \text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} - \text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\} \tag{17}$$

and

$$\begin{aligned}
 \rho\{Z(\mathbf{u}_i) \mid \lambda_{i-1}, \Lambda_{i-1}\} &= \frac{1}{2}[\text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} - \text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}] \\
 &\quad + \left[\left(\sqrt{\text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}} - \sqrt{\text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}} \right)^2 \right] \\
 &= \text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} \\
 &\quad \times \left(1.0 - \sqrt{\text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\} / \text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}} \right)
 \end{aligned} \tag{18}$$

Equation (18) indicates that SEA loss is determined from the two posterior variances: $\text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}$ and $\text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}$. Properties of SEA loss are detailed in Luo (1998), where SEA loss is shown to be positive, monotonically decreasing with regard to increasing data in the neighborhood, monotonically increasing with regard to increasing overall information and bounded between 0 and $\text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}$.

It is practical to consider the relative SEA loss or RSEA loss, defined as

$$\begin{aligned}
 \rho_{\text{R}}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}, \Lambda_{i-1}\} &= \rho\{Z(\mathbf{u}_i) \mid \lambda_{i-1}, \Lambda_{i-1}\} / \text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} \\
 &= 1.0 - \sqrt{\text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\} / \text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}} \tag{19}
 \end{aligned}$$

The RSEA loss is valued in the interval [0, 1]. A decreasing RSEA loss reflects a decreasing difference between two posterior variances. When the posterior variances become identical, the RSEA loss reaches zero. An advantage of the RSEA loss is that it is only affected by the grid size used, the range and type of the covariance model. The assessment, using RSEA loss, of the effects of different grid sizes and covariance models is examined next.

Assessment of the Relative SEA Loss

This section deals with the assessment of the upper bound of the RSEA loss for a given covariance model and grid ratio l/a , where a is the range and l is the lag (size) of the grid. Two common covariance models are first discussed in detail: the spherical and exponential models; the Gaussian covariance model is visited at the end of this section and provides a counterexample to the previous two.

For a given grid ratio, the RSEA loss reaches its upper bound when $\text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}$ reaches its lower bound and $\text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}$ reaches its upper bound simultaneously. The lower bound of $\text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\}$ is given by

$$\min_{\Lambda_{i-1}} \text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\} = \lim_{\Lambda_{i-1} \rightarrow \infty} \text{Var}\{Z(\mathbf{u}_i) \mid \Lambda_{i-1}\} = \text{Var}\{Z(\mathbf{u}_i) \mid (\infty)\}$$

and the upper bound of $\text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}$

$$\max_{\lambda_{i-1}} \text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} = \lim_{\lambda_{i-1} \rightarrow 1} \text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\} = \text{Var}\{Z(\mathbf{u}_i) \mid (1)\}$$

where $\text{Var}\{Z(\mathbf{u}_i) \mid (\infty)\}$ denotes the variance posterior to the information on the whole grid and $\text{Var}\{Z(\mathbf{u}_i) \mid (1)\}$ denotes the variance posterior to the closest datum. Then, the RSEA loss is

$$\rho_R\{Z(\mathbf{u}_i) \mid \lambda_{i-1}, \Lambda_{i-1}\} \leq \rho_R^{\text{upp}} = 1.0 - \sqrt{\text{Var}\{Z(\mathbf{u}_i) \mid (\infty)\} / \text{Var}\{Z(\mathbf{u}_i) \mid (1)\}}$$

$\text{Var}\{Z(\mathbf{u}_i) \mid (\infty)\}$ converges very fast for a given grid ratio,

$$\text{Var}\{Z(\mathbf{u}_i) \mid (\infty)\} \approx \text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}$$

where λ_{i-1} is a neighborhood with a considerably small size (e.g., 30). This can be demonstrated from the following configurations. $\text{Var}\{Z(\mathbf{u}_i) \mid \lambda_{i-1}\}$ is calculated by first considering the 8 closest locations of the grid, shown in Figure 3(a); then 24 locations, shown in Figure 3(b); then 48 locations, shown in Figure 3(c), and so on. The results of the spherical and exponential covariance models are shown in Figures 4 and 5, respectively, both indicating $\text{Var}\{Z(\mathbf{u}_i) \mid (\infty)\}$ can be approximated sufficiently by considering only the 24 closest locations of the grid

$$\text{Var}\{Z(\mathbf{u}_i) \mid (\infty)\} \approx \text{Var}\{Z(\mathbf{u}_i) \mid (24)\}$$

On the other hand, $\text{Var}\{Z(\mathbf{u}_i) \mid (1)\}$ is given by

$$\text{Var}\{Z(\mathbf{u}_i) \mid (1)\} = C(0) - C(h)^2 / C(0)$$

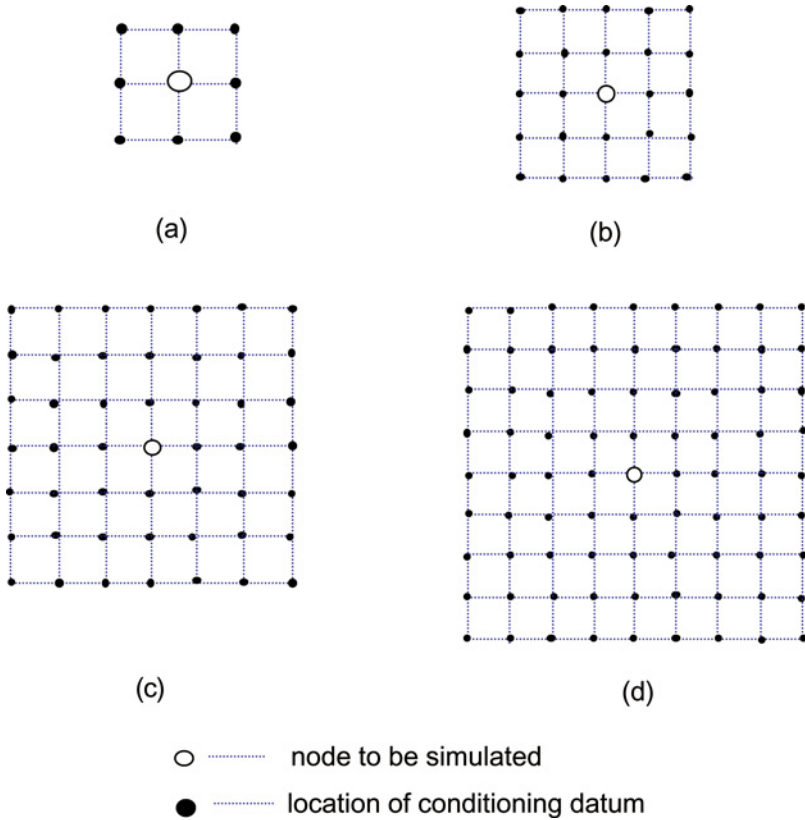


Figure 3. Four neighborhoods with regular grid: (a) 8, (b) 24, (c) 48, and (d) 80.

where $C(h)$ is the covariance and h denotes the distance of the closest datum. Figures 6 and 7 show the upper bounds of the exponential and spherical models, respectively, with the grid ratio changing from 0.001 to 0.5. The results indicate that (i) the upper bound of the RSEA loss is decreasing when the grid ratio is increasing and (ii) the upper bound is between 0.0 and 0.42.

Optimal Size of Neighborhood Using RSEA Loss

The RSEA loss in both the exponential model and the spherical model is, in practice, much smaller than its upper bound, and this makes the GSGS- ν effective. To show practical variations of the RSEA loss, the data configuration and the neighborhood are designed as follows. Randomly pick 120 nodes from a 21×21 grid around the node to be simulated. These nodes form the configuration

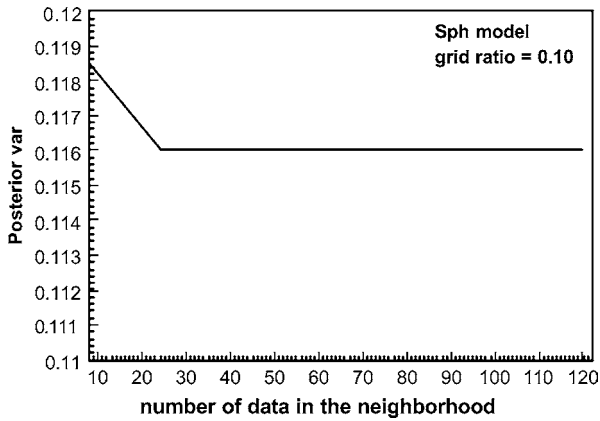


Figure 4. The convergence of the posterior variance of the spherical model, for the grid ratio of 0.1.

of the data, thus the simulated node ratio = $\frac{\text{simulated nodes}}{\text{all nodes}} \approx \frac{1}{4}$. Divide the area into four subareas, select the closest node in each subarea to construct a 4-location neighborhood, and calculate the RSEA loss. Then, choose the two closest nodes in each sub-area to construct an 8-location neighborhood, and calculate the RSEA loss. Keep extending the neighborhood to assess the variation of the RSEA loss. This process can be repeated several times to generate a reliable evaluation for the RSEA loss.

Figure 8 shows the results of the RSEA loss for the exponential model for grid ratio $l/a = 0.01, 0.05, \text{ and } 0.1$, with 50 repetitions. On the basis of these

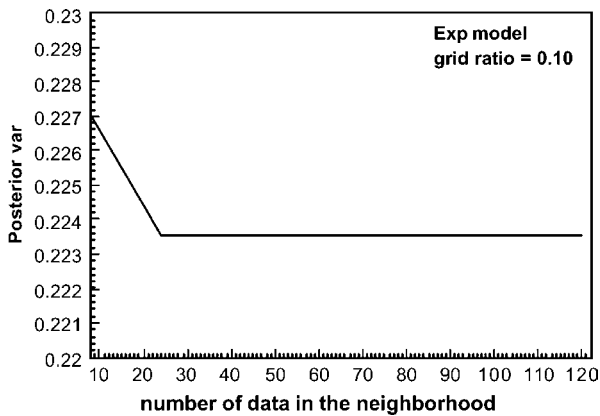


Figure 5. The convergence of the posterior variance of the exponential model, for the grid ratio of 0.1.

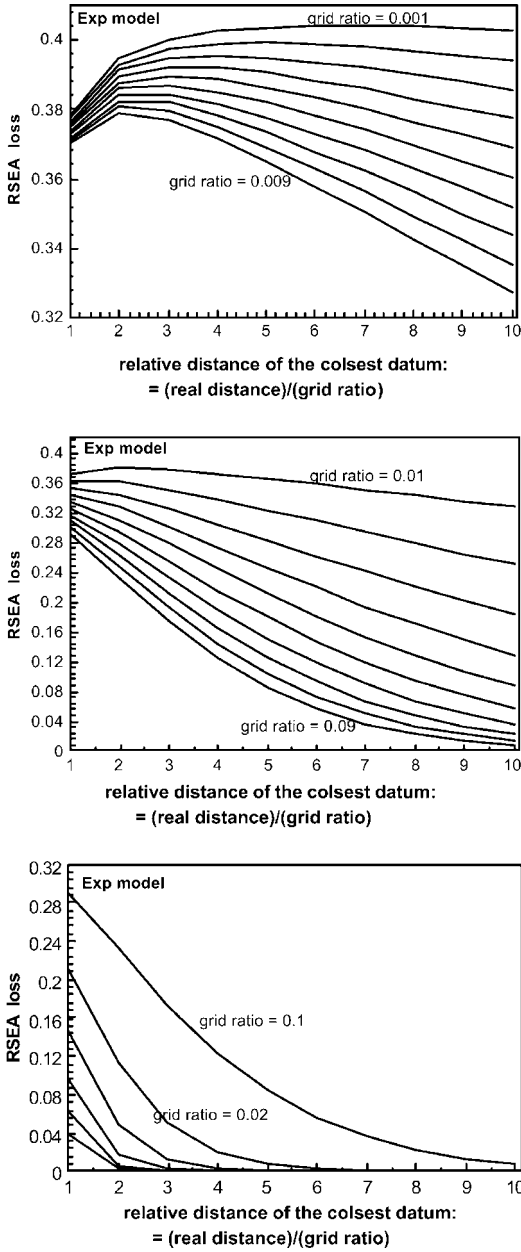


Figure 6. The upper bound of the RSEA loss of the exponential model with respect to different grid ratios and relative closest distances (=distance/grid ratio).

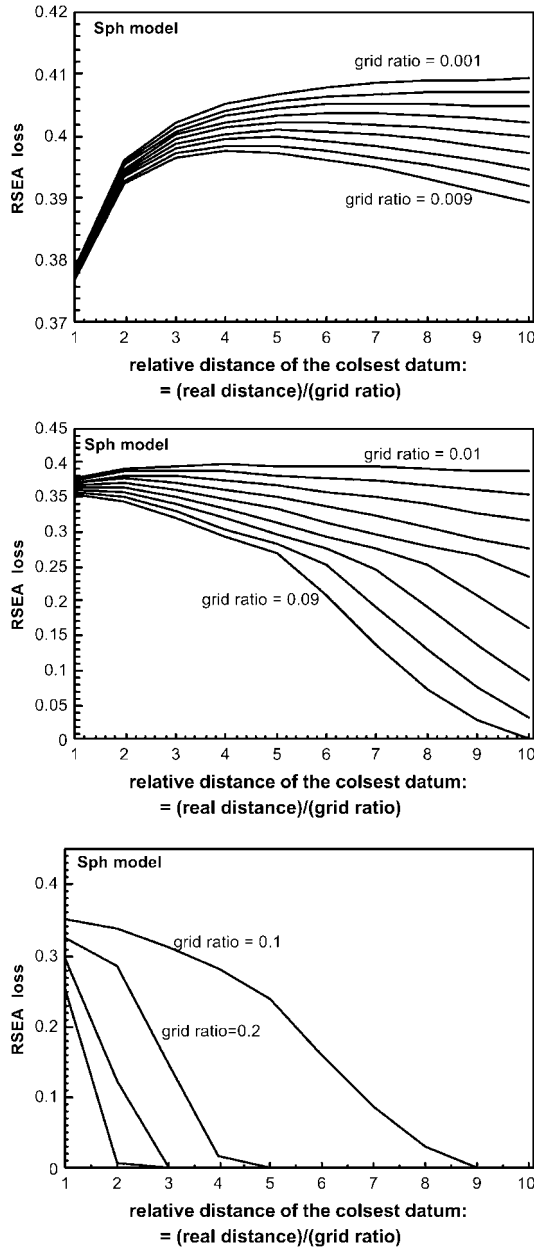


Figure 7. The upper bound of the RSEA loss of the spherical model with respect to different grid ratios and relative closest distances (=distance/grid ratio).

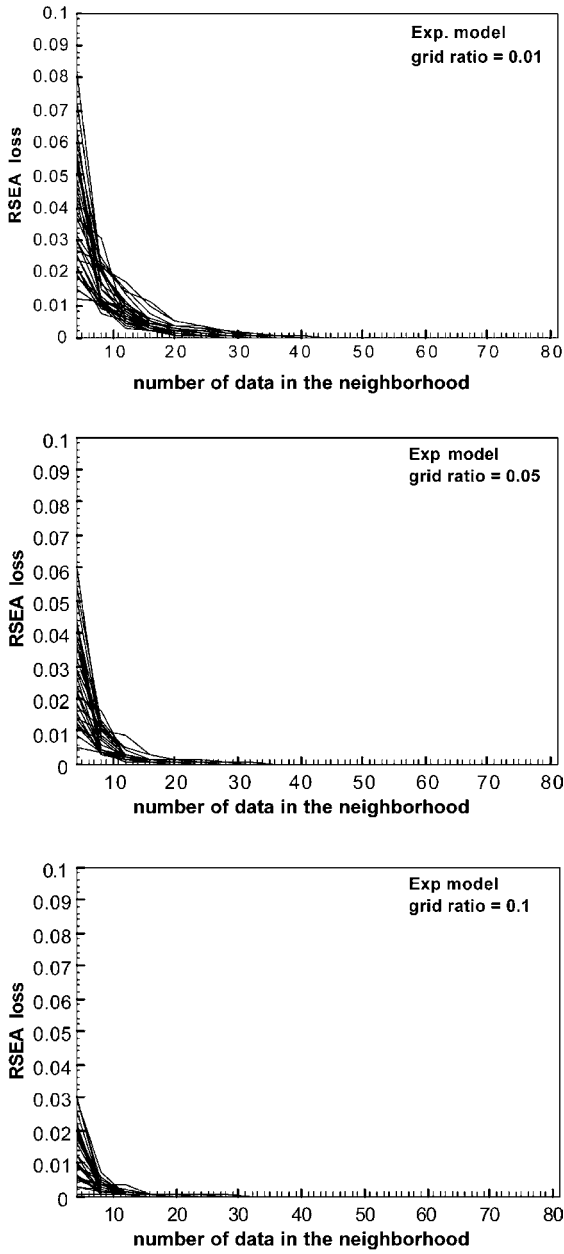


Figure 8. The RSEA loss of the exponential model for grid ratios of 0.01, 0.05, and 0.10. There are 120 “real” data randomly distributed in a grid of 21×21 .

Table 1. Optimal Size of the Neighborhood for the Exponential Model

Grid ratio (l/a)	Optimal size (RSEA loss <5%)	Optimal size (RSEA loss <1%)
≤ 0.01	2×4	5×4
> 0.01 and ≤ 0.1	2×4	4×4
> 0.1	1×4	3×4

results, the optimal size of the neighborhood of the exponential model with two RSEA loss requirements, less than 5% and less than 1%, is shown in Table 1. In general, the optimal size decreases when the grid ratio increases. When the grid ratio is larger than 0.1, and the RSEA loss requirement is less than 5%, it is enough to pick only one datum in each sub-area to construct a 4-location neighborhood.

The RSEA loss does not vary monotonically for the spherical model. This is demonstrated in Figures 9, 10, and 11. It is interesting that the RSEA loss is small and does not vary much when the grid ratio is no larger than 0.05. When the grid ratio is between 0.07 and 0.20, the RSEA loss increases considerably, and then it decreases rapidly when the grid ratio is over 0.25. The optimal size of the neighborhood with respect to different grid ratios is given in Table 2.

Both the exponential model and the spherical model show the property of the screen effect in the sense that a relatively small neighborhood can satisfy a less-than 5% RSEA loss requirement. This suggests the excellent performance of the GSGS- ν implementation.

The performance of the GSGS- ν for the exponential and spherical models does not imply that it is applicable for any covariance model. A counterexample is the Gaussian model. Figure 12 shows the RSEA loss of the Gaussian model for the grid ratios of 0.15, 0.20, and 0.25, indicating that the RSEA loss increases when the grid ratio decreases. When the grid ratio is less than 0.25, the RSEA loss is relatively high (much greater than 5%). This means that the GSGS- ν is much less effective for the Gaussian model in most applications.

The RSEA loss provides an assessment of the neighborhood size as a function of the variogram and the grid size. In practical terms, once the grid size is determined, a group size can be selected that minimizes computational costs without any substantial loss in accuracy.

CONCLUSIONS

The generalization of the sequential Gaussian simulation method introduced in this paper is termed the “generalized sequential Gaussian simulation on

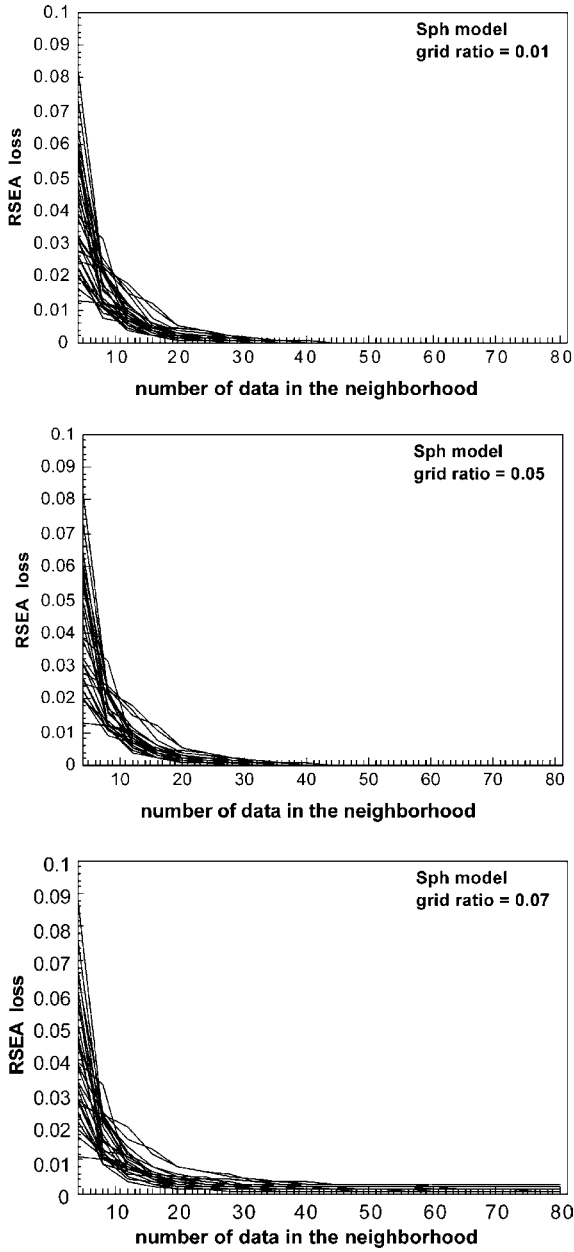


Figure 9. The RSEA loss of the spherical model for grid ratios of 0.01, 0.05, and 0.07. There are 120 “real” data randomly distributed in a grid of 21×21 .

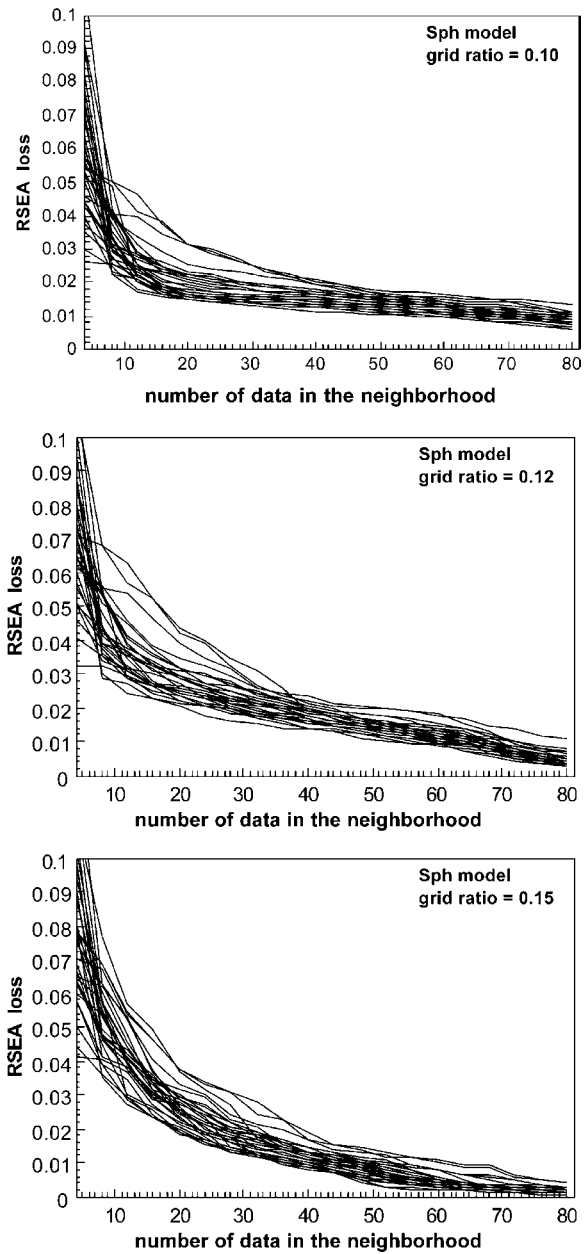


Figure 10. The RSEA loss of the spherical model for grid ratios of 0.10, 0.12, and 0.15. There are 120 “real” data randomly distributed in a grid of 21×21 .

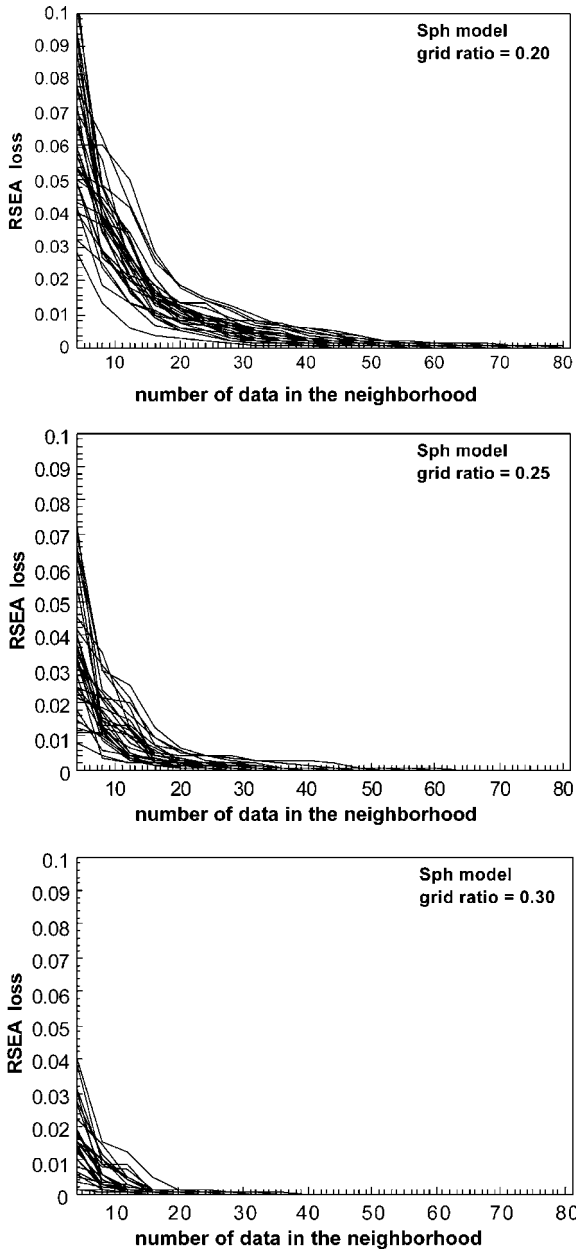


Figure 11. The RSEA loss of the spherical model for grid ratios of 0.20, 0.25, and 0.30. There are 120 “real” data randomly distributed in a grid of 21×21 .

Table 2. Optimal Size of the Neighborhood for the Spherical Model

Grid ratio (l/a)	Optimal size (RSEA loss <5%)	Optimal size (RSEA loss <1%)
≤ 0.07	2×4	6×4
> 0.07 and ≤ 0.17	5×4	Not available
> 0.17 and ≤ 0.25	5×4	9×4
> 0.25	2×4	5×4

group size ν " (GSGS- ν). The generalization takes advantage of the neighborhood sharing of adjacent nodes, simulating groups of nodes at a time, instead of the node-by-node simulation of SGS. This leads to sequential simulation implementations that are computationally efficient. Although implementations of GSGS are not presently available in the public domain, the method is relatively straightforward to implement.

GSGS- ν is based on the group decomposition of the posterior probability density function of a stationary Gaussian random field into posterior probability densities for a set of groups. The sequential Gaussian and LU decomposition simulation methods can be seen as the end members of GSGS- ν . When ν is 1, the method is identical to SGS. When ν includes all the nodes to be simulated, the method is identical to LU decomposition.

The results presented suggest that, in terms of computing cost, the optimal size ν of a group is about 80% of the optimal neighborhood used for SGS. The computational efficiencies are substantial.

The effectiveness of the GSGS- ν depends on the performance of SEA, characterized by the corresponding SEA loss. The SEA loss is a general measure of accuracy, defined as a mean-square difference between the simulated value posterior to the information in the neighborhood and the simulated value posterior to all information, and determined by the corresponding posterior variances. The relative SEA loss is a practical alternative as it is affected by the grid size used, the range and type of the covariance model. The results presented herein show that both the exponential and spherical covariance models can satisfy a less-than 5% RSEA loss requirement for any grid setup, using a relatively small neighborhood. However, the Gaussian covariance model was found to have a relatively high RSEA loss in most cases. It is suggested that, in most situations, the increase in speed in GSGS- ν is not at the expense of accuracy.

SEA loss is a general measure of accuracy, which may be used in any simulation method. Using this tool, the optimal size of a neighborhood that will satisfy an acceptable SEA loss for given conditions can be assessed before implementing the actual simulation.

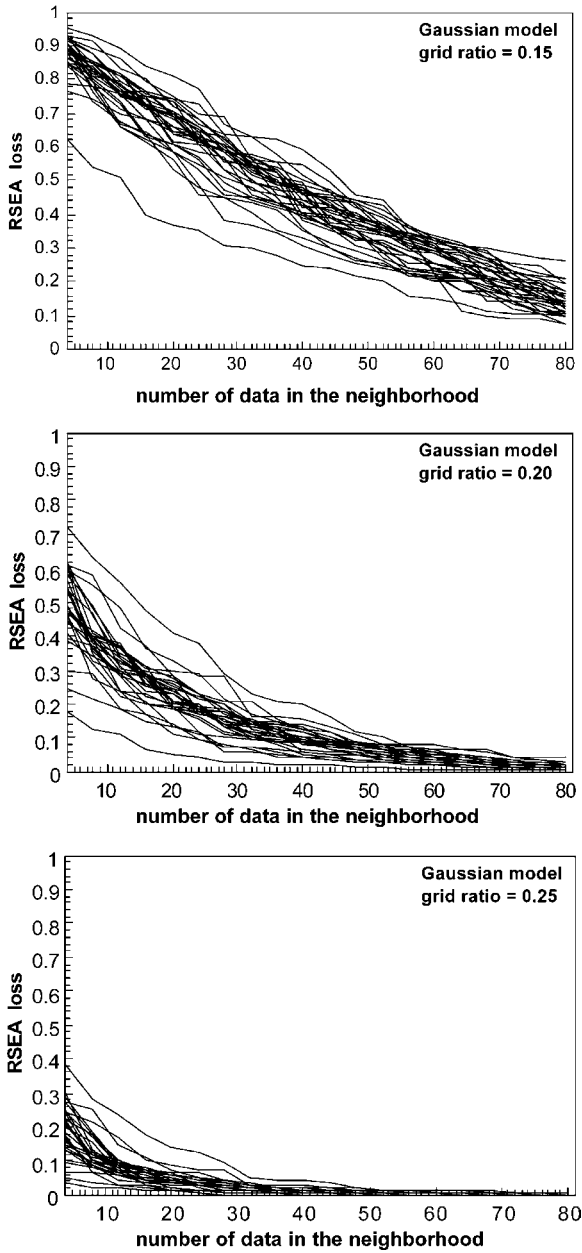


Figure 12. The RSEA loss of the Gaussian model for grid ratios of 0.15, 0.20, and 0.25. There are 120 “real” data randomly distributed in a grid of 21×21 .

ACKNOWLEDGMENTS

Funding was provided by the National Science and Engineering Research Council of Canada Grant No. OGP0105803 to R. Dimitrakopoulos. Thanks are to C. Dietrich, A. Boucher, and the reviewers of *Mathematical Geology* for their constructive comments and suggestions.

REFERENCES

- Alabert, F., 1987, Stochastic imaging of spatial distributions using hard and soft information: Unpublished MSc thesis, Stanford University, Stanford, CA, 197 p.
- Berger, J. O., 1985, Statistical decision theory and Bayesian analysis: Springer-Verlag, New York, 617 p.
- Cormen, T. H., Leiserson, C. E., and Rivest, R. L., 1990, Introduction to algorithms: The MIT Press, Cambridge, MA, 1028 p.
- David, M., 1977, Geostatistical ore reserve estimation: Elsevier, Amsterdam, 364 p.
- Davis, M. W., 1987a, Production of conditional simulations via the LU triangular decomposition of the covariance matrix: *Math. Geol.*, v. 19, no. 2, p. 91–98.
- Davis, M. W., 1987b, Generating large stochastic simulation—The matrix polynomial approximation method: *Math. Geol.*, v. 19, no. 2, p. 99–107.
- Dietrich, C. R., and Newsam, G. N., 1996, Fast and exact method for multidimensional Gaussian stochastic simulations: Extension to conditional simulations: *Water Resour. Res.*, v. 32, no. 6, p. 1643–1652.
- Dowd, P. A., and Saraç, C., 1994, An extension of the LU decomposition method of simulation, *in* Armstrong, M., and Dowd, P.A., eds., *Geostatistical simulations*: Kluwer, Dordrecht, p. 23–36.
- Godoy, M., 2003, The effective management of geological risk in long-term production scheduling of open pit mines: Unpublished PhD thesis, The University of Queensland, Brisbane, Qld., 256 p.
- Isaaks, E. H., 1990, The application of Monte Carlo methods to the analysis of spatially correlated data: Unpublished PhD thesis, Stanford University, Stanford, CA, 213 p.
- Johnson, M., 1987, Multivariate statistical simulation: Wiley, New York, 212 p.
- Journel, A.G., 1994, Modeling uncertainty: Some conceptual thoughts, *in* Dimitrakopoulos R., ed., *Geostatistics for the next century*: Kluwer, Dordrecht, p. 30–43.
- Luo, X., 1998, Spatiotemporal stochastic models for earth science and engineering applications: Unpublished PhD thesis, McGill University, Montreal, 177 p.
- Omre, H., Solna, K., and Tjelmeland, H., 1993, Simulation of random functions on large lattices, *in* Soares, A., ed., *Geostatistics Troia '92*: Kluwer Acrs, Dordrecht, p. 179–199.
- Ravenscroft, P. J., 1994, Conditional simulation for mining: Practical implementation in an industrial environment, *in* Armstrong, M., and Dowd, P. A., eds., *Geostatistical simulations*: Kluwers, Dordrecht, p. 79–87.
- Ripley, B. D., 1987, *Stochastic simulations*: Wiley, New York, 214 p.
- Rossenblatt, M., 1952, Remarks on multivariate transformation: *Ann. Math. Stat.*, v. 23, p. 470–472.
- Rubinstein, R. Y., 1981, *Simulation and the Monte Carlo method*: Wiley, New York, 278 p.
- Scheuer, E. M., and Stoller, D. S., 1962, On the generation of normal random vectors: *Technometrics*, v. 4, p. 278–281.
- Steinberg, D., 1974, *Computational matrix algebra*: McGraw-Hill, New York, 280 p.

APPENDIX: SOME PROPERTIES OF POSTERIOR COVARIANCES

The posterior covariance between $Z(\mathbf{u}_i)$ and $Z(\mathbf{u}_j)$, given (k) , is

$$C_{ij \cdot k} = \text{Cov}[Z(\mathbf{u}_i), Z(\mathbf{u}_j) \mid (k)] = C_{ij} - \mathbf{C}_{ik} \mathbf{C}_{kk}^{-1} \mathbf{C}_{kj}$$

The posterior covariance characterizes the correction of the prior covariance posterior to information (k) (e.g., Berger, 1985). For a positive definite covariance matrix the posterior covariance has the following properties.

Property 1. The covariance of $Z(\mathbf{u}_i)$ and $Z(\mathbf{u}_j)$ posterior to (k) , $C_{ij \cdot k} = \text{Cov}[Z(\mathbf{u}_i), Z(\mathbf{u}_j) \mid (k)]$, is equal to zero if (i) $Z(\mathbf{u}_i)$ has no correlation with $Z(\mathbf{u}_j)$ and the information (k) ; or (ii) $Z(\mathbf{u}_i) \subseteq (k)$ or $Z(\mathbf{u}_j) \subseteq (k)$.

Proof: (i) This is obvious from Equation (16):

$$C_{ij \cdot k} = C_{ij} - \mathbf{C}_{ik} \mathbf{C}_{kk}^{-1} \mathbf{C}_{kj} = 0 - \mathbf{0} \mathbf{C}_{kk}^{-1} \mathbf{C}_{kj} = 0$$

(ii) Assume $Z(\mathbf{u}_i) \subseteq (n)$, then vector $\mathbf{C}_{ik} = [\text{Cov}(Z(\mathbf{u}_i), Z(\mathbf{u}_1)), \dots, \text{Cov}(Z(\mathbf{u}_i), Z(\mathbf{u}_i)), \dots, \text{Cov}(Z(\mathbf{u}_i), Z(\mathbf{u}_k))]$ is but a row of the covariance matrix of (k) , \mathbf{C}_k , which entails

$$\mathbf{C}_{ik} \mathbf{C}_{kk}^{-1} = (0, \dots, 0, 1, 0, \dots, 0)$$

This is a row with all zeros except the i th element being one. Consequently,

$$\mathbf{C}_{ik} \mathbf{C}_{kk}^{-1} \mathbf{C}_{kj} = C_{ij}$$

Therefore the posterior covariance

$$C_{ij \cdot k} = C_{ij} - \mathbf{C}_{ik} \mathbf{C}_{kk}^{-1} \mathbf{C}_{kj} = 0$$

Property 2. The variance of $Z(\mathbf{u}_i)$ posterior to (p) is less than the variance of $Z(\mathbf{u}_i)$ posterior to (q) if $(p) \subseteq (q)$.