Some Geostatistical Software Implementation Details

Jared L. Deutsch and Clayton V. Deutsch

This paper considers four (mostly unrelated) problems in geostatistical software implementation, 1) duplicates and their removal, 2) sequential Gaussian simulation with a sill different from unity, 3) calculating a cross variogram for non-isotopic (non-collocated) data and 4) calculation of a global estimate (declustered mean) and global estimation variance. All of these software implementation details have been implemented in GSLIB compatible programs: `duplicates`, `sgsim_sill`, `gamv_nic` and `globvar`, respectively.

Multiple data values at the same location (duplicates) can cause a number of problems for geostatistical analysis. Prior to geostatistical analysis, duplicates should be removed, but this removal is nontrivial if the duplicated entries have different data values. Three methods for removing duplicates are considered; random removal of duplicated entries, using an average value or considering the spatial coherency of the duplicates.

Another software consideration arises in applying sequential Gaussian simulation with a variogram sill that is not unity. This is frequently necessary when simulating with a hole effect variogram or a domain with long range stationarity but trends at short range.

The calculation of cross variograms for non-isotopic data is not easy, as the cross covariance at h=0 cannot be calculated. This value is needed as the cross variogram is equal to the cross covariance at h=0 minus the covariance function. The cross covariance at the origin is calculated by fitting a spherical covariance model to known cross covariance values and extrapolating to h=0.

A global estimate and estimation variance is a useful statistic for reporting the mean and uncertainty of a deposit. The calculation of these values using global kriging of the entire domain has been streamlined in a GSLIB-like program.

1. Introduction

This paper considers four problems in geostatistical software implementation, 1) duplicate data finding and removal, 2) sequential Gaussian simulation with a sill not equal to one, 3) calculating a cross variogram with non-isotopic data and 4) calculation of a global estimate and global estimation variance.

Three strategies for duplicate removal including random removal, averaging or picking the most spatially coherent value are implemented in the GSLIB compatible program `duplicates`. These strategies are tested using a very large data set containing a number of duplicate values.

The GSLIB program `sgsim` was modified to include the option to specify a sill value for simulating with a sill value that is not unity. Issues relating to sequential Gaussian simulation with different sill values are discussed as well as variogram and histogram reproduction.

The extrapolation of a cross covariance at h=0 for flipping a cross covariance of non-isotopic data is implemented in a modified version of `gamv`. This is a necessary step for modeling for the cross variogram subject to the constraints of the linear model of coregionalization.

The calculation of a global estimate (equivalently a declustered mean) and global estimation variance with kriging is a useful exercise, but not easily implemented in existing programs. Global kriging over a specified domain has been implemented in the GSLIB-like program `globvar`.

2.1. Duplicate Removal

Consider a data set with multiple data points at the same location. These duplicate data points can cause numerous problems when analyzing the data. The first major problem is that duplicate values can skew summary statistics. This problem can be alleviated by first using cell declustering to reduce the weight of duplicates; however, kriging with duplicates generates singular matrices. This results in unestimated regions near the duplicate locations. Prior to kriging and analysis, duplicate entries in data should be removed to prevent this.

The removal of duplicates that have the same value is trivial, duplicated data points can be removed leaving only a single entry. However, if the duplicate entries have different values, possibly stemming from multiple measurements or sources, then this removal is nontrivial. We propose three methods for dealing with duplicates which are implemented in the GSLIB compatible program `duplicates`.

The first method is to randomly choose a duplicate to keep and eliminate the others. Randomly choosing a duplicate value does not make assumptions about how the duplicate values were generated or about the nature of the data value.

A second method to deal with duplicates is to average the duplicated values. This assumes that the duplicates are multiple measurements at the same location and that an arithmetic average is the most appropriate method for combining these measurements. This situation could arise if the same measurements were taken on core multiple times by the same lab or core data from multiple labs was combined.

The third method proposed is to consider the spatial coherency of the duplicate values. A local search for data near the duplicates is done to calculate a local average. The duplicate value closest to the average is kept and the other entries discarded. If multiple duplicates are equidistant from the local average then the elimination is done randomly. If the duplicated data may be the result of corrupted data, then this strategy is recommended.

2.2. Implementation

The standalone program `duplicates` implements the duplicate removal methods described. Duplicate finding is implemented for up to 4 dimensions (Deutsch *et al*., 2010) with specified tolerances. Trimming limits can be specified; any data values falling outside of the trimming limits will be reset to -999. The parameters for `derror` are:

```
Line
1                       Parameters for DUPLICATES
2                       *************************
3       START OF PARAMETERS:
4       data.dat                    -file with data
5       2   3   0   0   4           -  columns for x,y,z,d,var
6       -1.0e21   1.0e21            -  trimming limits
7       0                           -use same dmin for all dimensions?
8       3.5   2.0   0.0   0.0       -  dmin for x,y,z,d
9       1    1.e3                   -duplicates (option),srchscl
10      data_dup.out                -file for output
11      data_sum.out                -file for summary of duplicates
12      69369                       -random number seed for pick random
```

The parameter file for `duplicates` is in the same style as many of the GSLIB programs. **Line 4** specifies the location of the Geo-EAS compatible data file (Deutsch and Journel, 1998). Column numbers for relevant variables are given in **Line 5**. The data can be trimmed (**Line 6**). The option is given to consider the same search radius for duplicates (**Line 7**) in which case the *x* search radius will be considered or different tolerances for each dimension (**Line 8**). The duplicate removal option is specified in **Line 9**, option 1 is random removal, option 2 averages and duplicates and option 3 looks for spatial coherency. The local search for option 3 considers points in a scaled up tolerance window. This scaling factor is specified in **Line 9**. The output file and summary file providing a list of duplicates found and action taken are specified in **Lines 10** and **11,** respectively. A random number seed must also be provided (**Line 12**) for the `acorni` random number generator (Deutsch and Journel, 1998).

2.3. Testing

A data set supplied by the Alberta Energy and Utilities Board (Warren, 2003) containing well picks from the Athabasca oil sands was searched for duplicates. This data set was supplied by numerous corporations (BP, Encana, CNRL, Petro-Canada to name a few) and contained a large number of duplicates. Many of these came from different companies that had collected data from the same wells. For testing purposes, net interval thickness was considered. A location map of this data is shown in Figure 1.

Since the duplicates appear to be the result of numerous laboratory analyses, the duplicates were averaged. A portion of the summary file output by `duplicates` is shown:

```
Line
1       ----------New Duplicate---------
2       ****Data kept:  6155917.00  523401.81
3           old value:     17.0900
4       ****new value:     17.0950
5       Data removed:  6155917.00  523401.81
6         data value:     17.1000
```

For each duplicate found, the original data values and locations are shown as well as the new value used for that location. In this case the average value was used. The resulting cleaned data set was kriged (Figure 2) using simple kriging with the corrected mean. Kriging was also done using random removal and the spatial coherency option; differences between these kriged maps and the average value were minor for this case study, so are not shown here. Kriging without removing duplicates resulted in an unestimated grid.

3.1. Sequential Gaussian Simulation with a Sill not-equal-to 1

Sequential Gaussian simulation (SGS) as implemented in `sgsim` is performed using a standardized sill of 1. This forces the input variogram to conform to the linear model of regionalization, in which the observed variable *Y* is a function of *n* standard normal factors Y_i (mean=0, variance=1) which each contribute $a_i^2=C_i$ to the variance (Equations 1-2).

$$Y = \sum_{i=0}^{n_{fac}} a_i Y_i \tag{1}$$

$$Var\{Y\} = \sum_{i=0}^{n_{fac}} (a_i)^2 = \sum_{i=0}^{n_{fac}} C_i \tag{2}$$

While SGS is generally performed with a standardized sill (therefore conforming to the linear model of regionalization), there are instances when simulation with a different sill would be desirable. For cyclic features where the use of a hole effect model is warranted, changing the sill value is useful to aid in fitting the experimental variogram. Short range trends in domains with long range stationarity can be modeled using a variogram that goes above the sill value. Both of these cases motivated a version of `sgsim` aimed at simulating variograms with a specified sill value.

3.2. Implementation

In SGS, a conditional mean is calculated based on the covariance between previously simulated nodes (or data values) and the location being simulated (further details in Deutsch and Journel, 1998). This calculation of the covariance between two locations separated by a lag vector **h** is now calculated as:

$$C(\mathbf{h}) = \text{Sill} - \sum_{i=0}^{nst} C_i + \sum_{i=0}^{nst} C_i \bullet \Gamma_i(\mathbf{h}) \tag{3}$$

where C_i is the variance contribution and Γ_i is the structured portion of the variogram. The only change to the `sgsim` parameters is the addition of a sill parameter when specifying the variogram. The sill value is specified on **Line 1** in the portion of the `sgsim_sill` parameter file shown:

```
Line
1      2    0.0  1.0                  -nst, nugget effect, sill
2      1    0.3  0.0   0.0   0.0     -it,cc,ang1,ang2,ang3
3             1500.0 1500.0  10.0    -a_hmax, a_hmin, a_vert
4      1    0.7  0.0   0.0   0.0     -it,cc,ang1,ang2,ang3
5            20000.0 5000.0  10.0    -a_hmax, a_hmin, a_vert
```

3.3. Case Study

Consider the unconditional simulation of a 10,000m x 10,000m 2D grid using SGS with 100 nodes in each direction. Simulation was done using a multiple grid search and a large number of previously simulated grid nodes to improve variogram reproduction. This resulted in lengthy run times of approximately 45 seconds on a dual core 3.2 GHz computer but better input reproduction. Three different variogram models were considered in this case study: a regular variogram modeled to a sill of 1, a variogram with a short range trend (total covariance of 1.3) and a hole effect model with a total covariance of 0.8 (shown in Figures 3 through 5).

The first variogram model was a regular variogram model with a sill of 1 with a measure of zonal anisotropy (Equation 4). Histogram reproduction was good (Figure 3); 100 simulations yielded a mean of 0.03 and standard deviation of 0.97 which are close to the expected 0.00 and 1.00, respectively. Variogram reproduction at all ranges was reasonable; however, variograms were accurately reproduced at short ranges (up to 2000m).

$$\gamma_1(\mathbf{h}) = 0.4Sph(\mathbf{h})_{\substack{ah1=2000 \\ ah2=1500}} + 0.6Sph(\mathbf{h})_{\substack{ah1=10000 \\ ah2=5000}} \quad (4)$$

The second variogram model considered a short range trend with the structured portion of the variogram accounting for a covariance of 1.3 (Equation 5). The histogram of simulated values produced a mean of 0.03 and standard deviation of 1.06 (Figure 4). The resulting distribution followed the standard Gaussian distribution. Variogram reproduction up to 4000m was reasonable; at distances greater than 4000m the variogram did not fully reach the sill of 1.3. This could be improved using an even larger number of previously simulated grid nodes for the simulation.

$$\gamma_2(\mathbf{h}) = 0.4Sph(\mathbf{h})_{\substack{ah1=1500 \\ ah2=1500}} + 0.9Sph(\mathbf{h})_{\substack{ah1=5000 \\ ah2=5000}} \quad (5)$$

A hole effect variogram was also considered with a structured covariance of 0.8. One direction exhibited cyclic behavior while the other direction had a large degree of zonal anisotropy (Equation 6). As with the prior examples, histogram reproduction was good (mean of 0.04 and standard deviation of 1.01, Figure 5). Variogram reproduction was reasonable up to 3500m; beyond this range the simulated variogram did not dip below the sill. This is likely a product of not using enough simulated grid nodes. This could also be improved by simulating a larger grid.

$$\gamma_3(\mathbf{h}) = 0.4Sph(\mathbf{h})_{\substack{ah1=1000 \\ ah2=1000}} + 0.4Hole(\mathbf{h})_{\substack{ah1=10000 \\ ah2=2500}} \quad (6)$$

4.1. Variogram calculation with non-collocated data

Cokriging requires a cross covariance model valid for all distances *h*. Consider the problem of calculating a cross covariance at *h*=0 for non-collocated data for calculating a cross variogram. While solutions using nearby data have been proposed (Wawruch et al., 2002), these rely on a large amount of user input to produce reasonable results. For this reason, a tool for calculating a cross covariance at *h*=0 to flip a cross covariance plot has been implemented in `gamv_nic`. Note that the calculated cross covariance must conform to the linear model of coregionalization (LMC). The LMC constraint means that the covariance matrix must be positive definite (ie: determinants of the covariance matrix must be positive). It is up to the user to check that the cross covariance at *h*=0 satisfies the LMC constraint. For 2 variables this means that Equation 7 must be satisfied for all structures where $C^{(i)}$ are the variance contributions (not the covariances).

$$\begin{aligned} & C_{YY}^{(i)} \geq 0 \\ & C_{ZZ}^{(i)} \geq 0 \qquad \text{for } i = 0,\ldots,nst \\ & C_{YY}^{(i)} C_{ZZ}^{(i)} \geq (C_{YZ}^{(i)})^2 \end{aligned} \quad (7)$$

To calculate the cross covariance at *h*=0, a spherical covariance model is fit to the cross covariance values using an iterative method. This spherical covariance fit is shown schematically in Figure 6. As data close to *h*=0 is more valuable, and the reliability of the calculated cross covariance is generally proportional to the number of data, the user can specify whether to use inverse distance weighting and/or number of pairs weighting. The objective function fit is given in Equation 8 if inverse distance weighting and number of pairs weighting are considered. This function is minimized by optimizing the fit of the spherical covariance function to the data.

$$Obj = \sum_{i=1}^{N} \left(\frac{1}{h} \right) \left(n_{pairs} \right) \left(C_{fit}(h) - C_{YZ}(h) \right)^2 \quad (8)$$

From the spherical covariance function, the cross covariance at *h*=0 can be calculated and used to flip the cross covariance values and calculate the cross variogram $\gamma_{YZ}(h) = C_{YZ}(0) - C_{YZ}(h)$.

4.2. Implementation

The parameter file for `gamv_nic` is very similar to `gamv`, with the exception that when modeling the cross covariance (option -3), the user must specify whether to use inverse distance weighting, number of points weighting and the option to add a nugget effect (**Line 1**). If you know what covariance value to use (option -30) you can specify this value (**Line 2**). These parameters are described in the parameter file in the accompanying electronic documentation.

```
Line
1       1   2  -3   1   1   0.10          -tail var., head var., varg. type
2       1   2  -30  0.90                  -tail var., head var., varg. type
```

4.3. Case Study

To test the reproduction of the flipped cross covariance, a two dimensional 50m x 50m grid was simulated with two standard normal (mean = 0, variance = 1) collocated variables with a correlation of 0.6. The parameters for the variograms are given in Equations 9 and 10. The resulting simulated correlation was 0.58, close the input 0.6. The simulated variables and data are included in the electronic documentation accompanying this paper.

$$\gamma_1(\mathbf{h}) = 0.1 + 0.9Sph(\mathbf{h})_{\substack{ah1=50 \\ ah2=50}} \tag{9}$$

$$\gamma_2(\mathbf{h}) = 0.15 + 0.85Sph(\mathbf{h})_{\substack{ah1=10 \\ ah2=10}} \tag{10}$$

From this 50x50 grid, samples for variable 1 and 2 were taken at 4m intervals, offset by 2m so that there was no collocated data. Variograms and covariances for each variable were calculated (Figure 7) using the simulated grids (solid lines) and non-isotopic samples (dashed lines). The flipped cross covariance was calculated using `gamv_nic` using inverse distance and number of points weighting with a 10% nugget effect. The flipped cross covariance was close to the true flipped cross covariance as shown in Figure 8. The calculated cross covariance at *h*=0 was 0.56, which is close to the true value of 0.58 which corresponds to the sill of the cross variogram.

5.1. Calculation of a Global Estimate and Global Estimation Variance

Consider the problem of calculating a global estimate and estimation variance using kriging. These values are useful as they effectively represent a declustered global mean and variance if the domain can be considered stationary. Using all data points α, kriging weights λ can be calculated (Equation 11). These weights can be used to calculate an estimate of the mean (Equation 12) and the average covariance of the domain with itself can be used to calculate the global estimation variance (Equation 13).

$$\left[\mathbf{C}(\alpha,\beta)\right]\left[\boldsymbol{\lambda}\right] = \left[\mathbf{C}(\alpha,A)\right] \quad \text{where } \alpha = \alpha_1,\ldots,\alpha_n \tag{11}$$

$$\overline{z} = \sum_{\alpha=1}^{n} \lambda_\alpha \tag{12}$$

$$\sigma^2_{E,\overline{z}} = \overline{C}(A,A) - 2\sum_{\alpha=1}^{n} \lambda_\alpha \cdot \overline{C}(A,A) + \sum_{\alpha=1}^{n}\sum_{\beta=1}^{n} \lambda_\alpha \lambda_\beta \cdot C(\alpha,\beta) \tag{13}$$

The importance of stationarity for the choice of domain *A* cannot be stressed enough. As the domain size A increase, the estimation variance will decrease, but this is only realistic if the domain chosen is stationary.

5.2. Implementation

The calculation of the global mean and estimation variance has been implemented in the GSLIB-like program `globvar`. The data file containing the location coordinates and variable of interest is specified in **Line 4** and column numbers in **Line 5**. The variable can be trimmed (**Line 6**). To specify the domain, a gridded file specifying the area of interest must be specified (**Line 7**). When reading this gridded file, a node with a value of 1 is taken to be in the domain and a value of 0 removes the node from the domain. The column for this series is specified in **Line 8** along with an option for the calculation of $\overline{C}(A,A)$. The calculation of $\overline{C}(A,A)$ is typically done by calculating the covariance between each domain point and every other domain point meaning that the number of calculations increases with the square of the number of points in the domain. If there are more than 5000 grid blocks, there is a huge number of precision errors associated with this calculation so the authors recommend that 25,000 location pairs be randomly chosen from within the domain. To calculate the covariance using all of the blocks, the option parameter is 0, otherwise 25,000 randomly chosen locations are used. The grid for the domain data is specified in standard GSLIB format (**Lines 9-11**). Finally, the variogram to be used is specified in standard GSLIB format following (**Lines 12-16**).

```
Line
1                       Parameters for GLOBVAR
2                       **********************
3     START OF PARAMETERS:
4     2dwelldata.dat                  -file with data
5     2  3  0  4  0                   -   columns for X,Y,Z,var,d
6     -1.0e21   1.0e21                -   trimming limits
7     domain.dat                      -gridded file with domain data
8     1    0                          -   column for ind and option for Cbar(A,A)
9     65   80.0    160.0              -   nx,xmn,xsiz
10    65   80.0    160.0              -   ny,ymn,ysiz
11    1    0.5    1.0                -   nz,zmn,zsiz
12    2    0.0                       -nst, nugget effect
13    1    0.3  0.0   0.0   0.0       -it,cc,ang1,ang2,ang3
14          1500.0  1500.0  10.0      -a_hmax, a_hmin, a_vert
15    1    0.7  0.0   0.0   0.0       -it,cc,ang1,ang2,ang3
16        20000.0  20000.0  10.0      -a_hmax, a_hmin, a_vert
```

5.3. Case Study

The global estimate and estimation variance was calculated for the familiar 2DWellData.dat (Figure 8). For this small case study the entire square domain shown in Figure 8 was used with a square grid of 65 160m x 160m blocks. The parameters used are included in the accompanying electronic documentation. The global estimate and variance were calculated using both options for $\bar{C}(A,A)$; resulting estimates are shown in Table 1. Note that the variance of the data was 3.6200. The authors recommend that the option for randomly choosing points for global kriging be used for models with greater than 5000 blocks to reduce the run time and precision errors. In this case study, randomly choosing 25,000 block pairs produced a global estimation variance very close to considering the 17,850,625 block pairs calculated if all nodes are used.

Table 1: Global estimate and global estimation variance using

	From Data Only	Global kriging with all points used for $\bar{C}(A,A)$	Global kriging with randomly chosen points for $\bar{C}(A,A)$
Global Estimate	8.4020	8.1486	8.1486
Global Estimation Variance	-	0.4306	0.4299

6. Conclusions

This paper addressed four issues in geostatistical software implementation including duplicate removal, implemented in the GSLIB compatible program `duplicates`, sequential Gaussian simulation with a sill value not equal to unity with `sgsim_sill`, a tool for non-isotopic sampling with `gamv_nic` and calculation of a global estimate and variance with `globvar`.

The duplicate removal software offers a number of options for duplicate removal; the practitioner should use their discretion to decide which method is appropriate but some basic guidelines were given. The output list of duplicates should aid in this decision.

Sequential Gaussian simulation with a sill not equal to 1 has been implemented, although the user is cautioned to check variogram reproduction. Variogram reproduction can be improved by using multiple grid searches and a large number of previously simulated nodes.

Geostatistical analysis of non-collocated data is difficult; a tool for flipping a cross covariance plot to calculate a cross variogram has been implemented to aid in this process. For the case study presented, fitting a spherical covariance function and extrapolating a cross covariance value at *h*=0 worked well using inverse distance and number of pairs weighting.

It is useful to have a global estimate and global estimation variance when analyzing a deposit. A tool for calculating these values has been implemented to aid the geostatistician in summarizing results. This calculation is done using global kriging over a specified domain in up to 4 dimensions.

References

Deutsch, C.V. and Journel, A.G., 1998, *GSLIB: Geostatistical Software Library and User's Guide*, Oxford University Press, New York, 2nd Ed., 369 pp.

Deutsch, J.L., Boisvert, J.B. and Deutsch, C.V., 2010, A New Dimension to Account for Data Error and Volume Support, *12th Annual Report of the Centre for Computational Geostatistics*, Paper 308.

Warren, A., 2003, Report 2003-A: EUB Athabasca Wabiskaw-McMurray Regional Geological Study, Alberta Energy and Utilities Board, 187 pp.

Wawruch, T.M., Deutsch, C.V. and McLennan, J.A., 2002, Geostatistical Analysis of Multiple Data Types that are not Available at the Same Locations, *4th Annual Report of the Centre for Computational Geostatistics*, Paper 33.

Figures

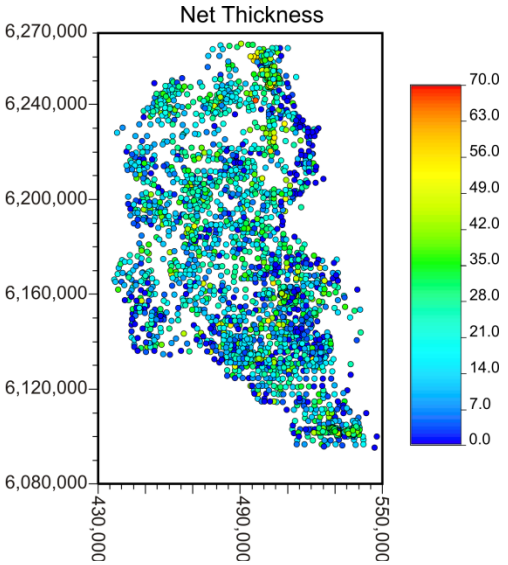

Figure 1: Locations of well picks from EUB data.

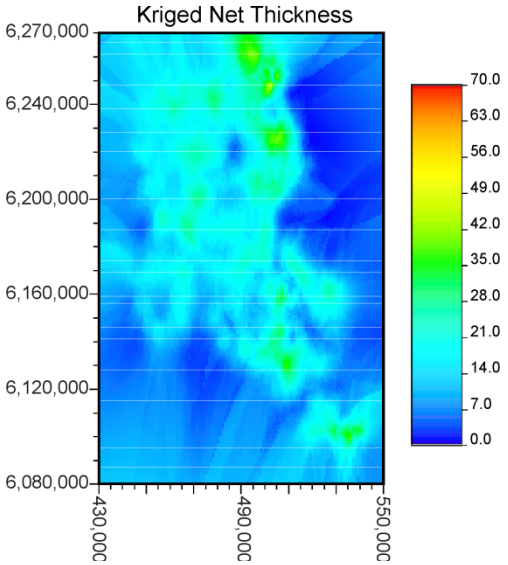

Figure 2: Kriged well picks with duplicates removed.

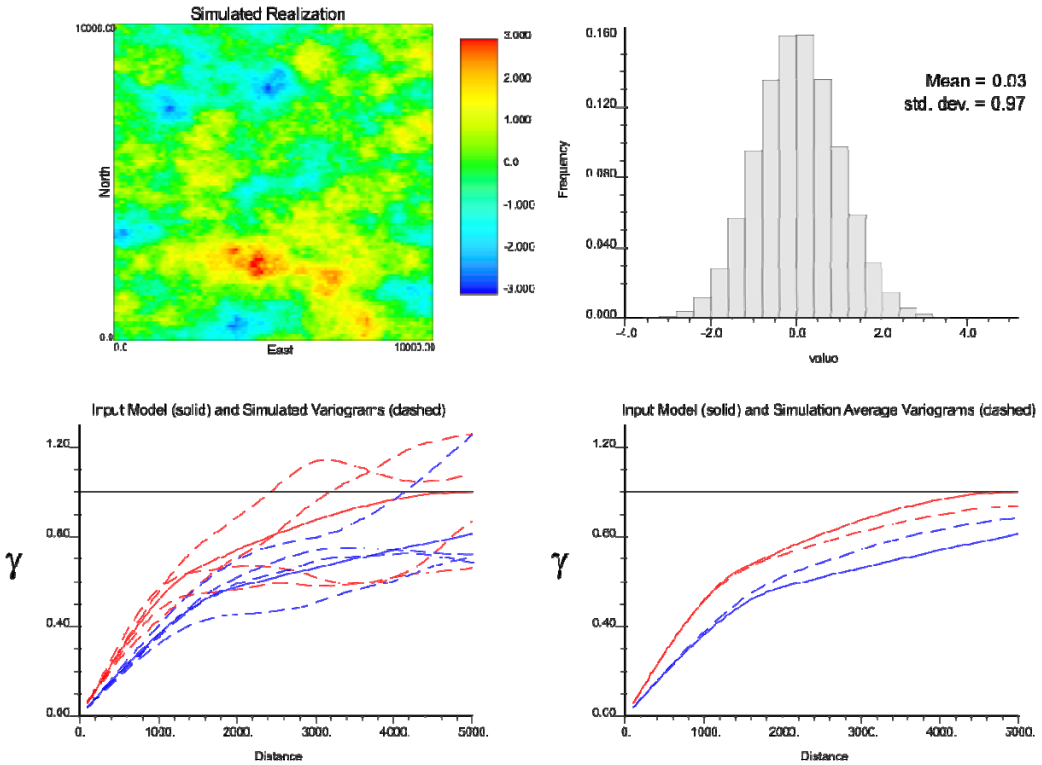

Figure 3: Example realization, histogram of simulated results and average variogram reproduction for a regular variogram model that reaches the sill of 1 (Equation 5).

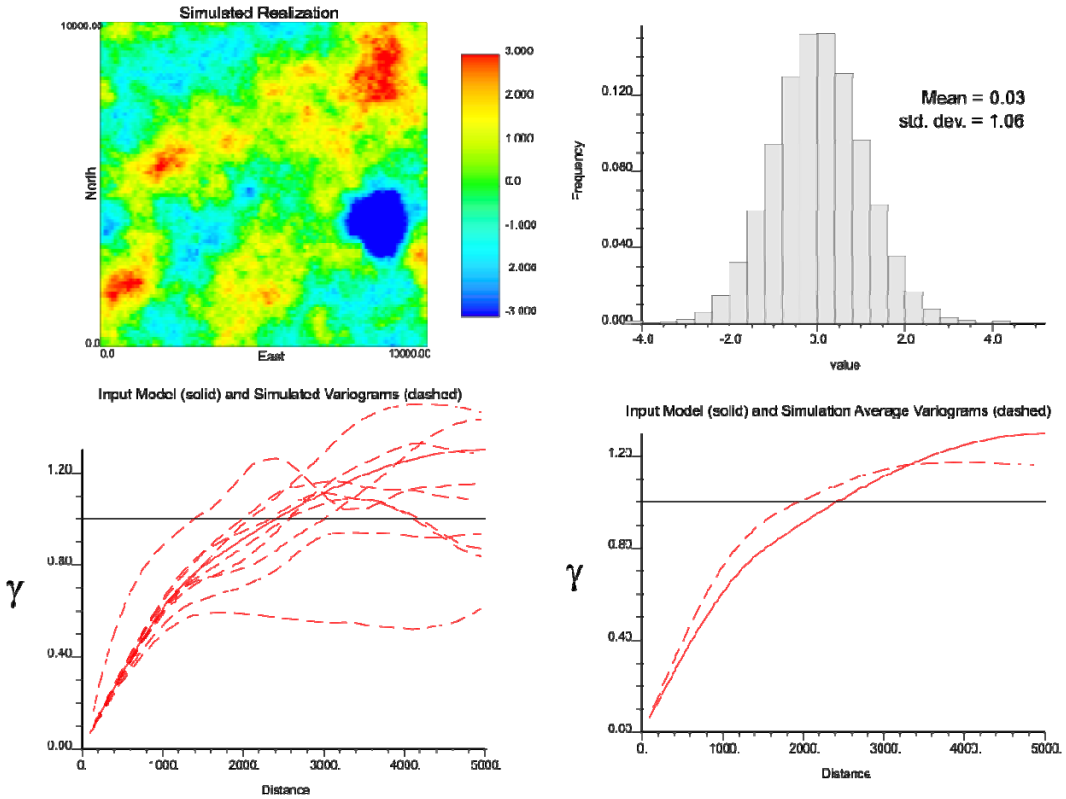

Figure 4: One realization, histogram of results and average variogram reproduction for a variogram model with a short range trend (Equation 6).

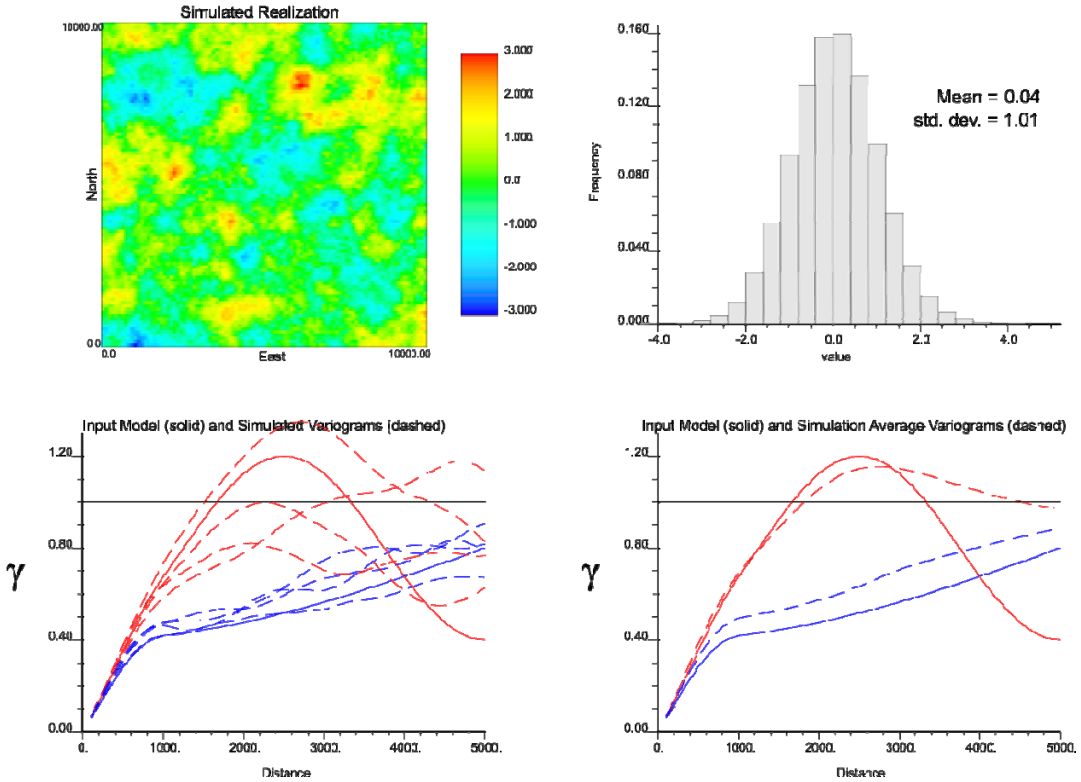

Figure 5: Example realization, histogram of resulting realizations and average variogram reproduction for a hole effect variogram model (Equation 7).

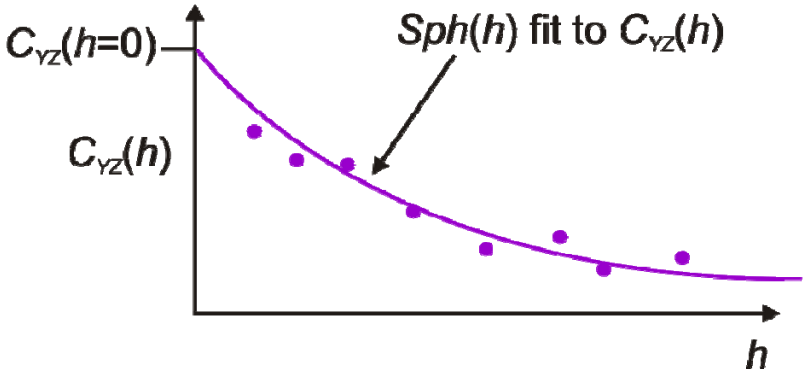

Figure 6: Fitting of cross covariance using a spherical covariance function to determine the cross covariance at h=0.

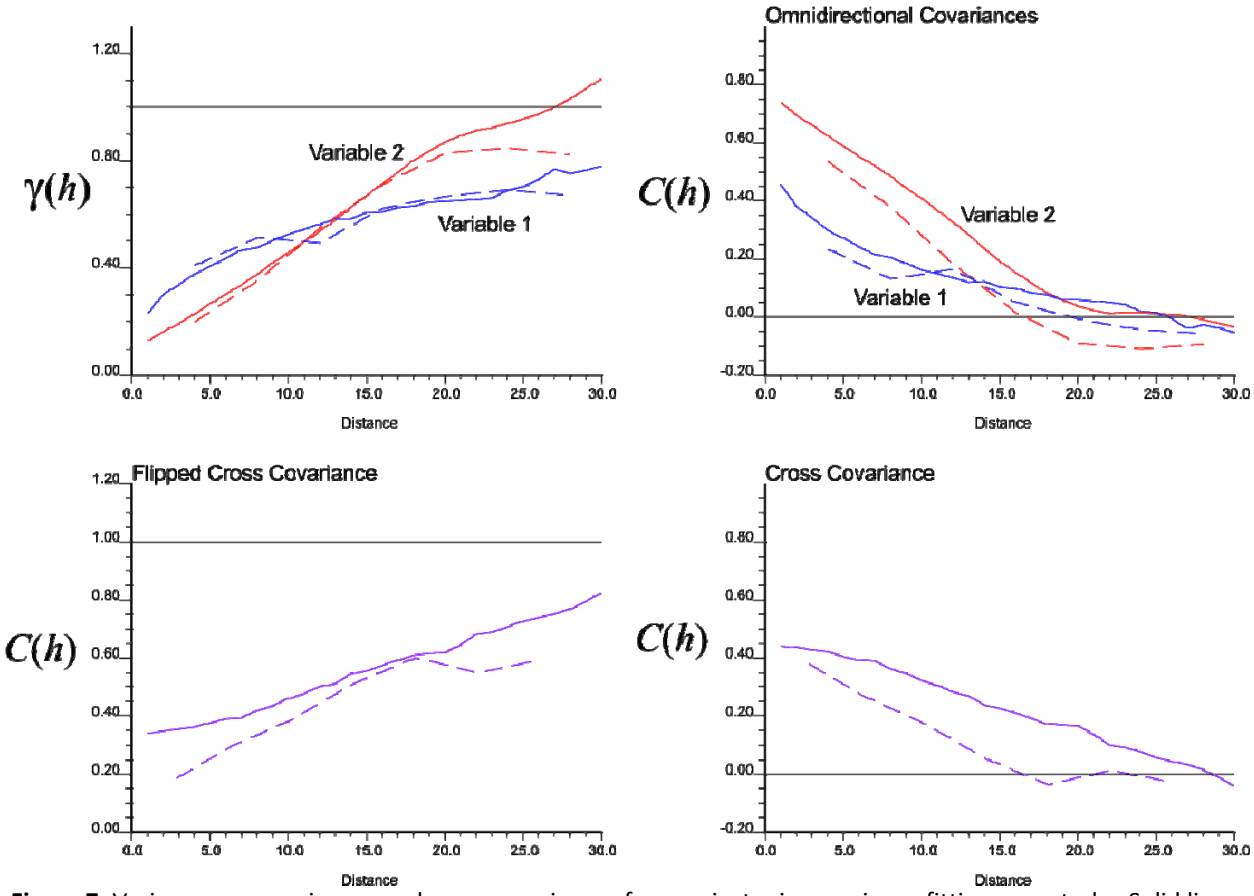

Figure 7: Variograms, covariances and cross covariances for non-isotopic covariance fitting case study. Solid lines were calculated using the true data, dashed lines using the sample values.

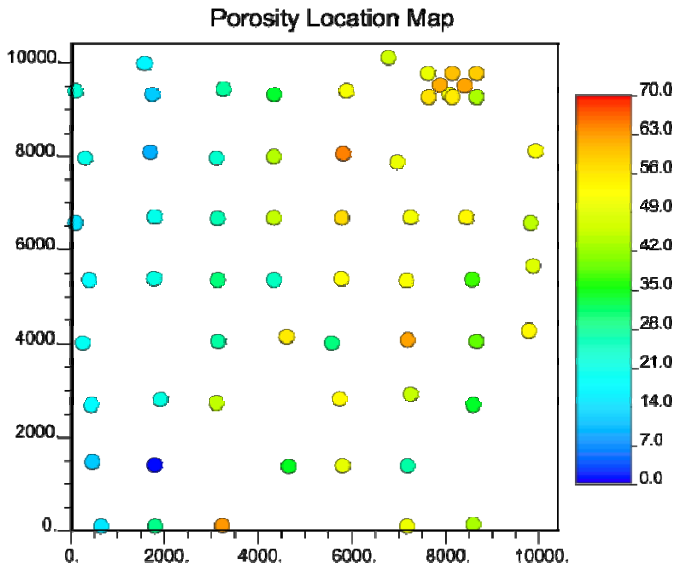

Figure 8: Location map of 2DWellData.dat. The entire square grid shown was considered to be the domain for this small case study.