Multivariate Block LU Simulation

Jeff Boisvert and Clayton V. Deutsch

Centre for Computational Geostatistics
Department of Civil & Environmental Engineering
University of Alberta

Generating recoverable reserves is a critical step in mining geostatistics. A methodology to generate recoverable reserves at large panel and selective mining unit (SMU) scale will be discussed. This paper is an extension of paper number 304 – Local Recoverable Reserves Prediction with Block LU Simulation. The software presented will expand the program, BLUSIM, to include cosimulation of multiple variables. The LU method of simulation allows for fast simultaneous generation of realizations for reserve calculation. An LMC model for the variables is required for the multivariate version of BLUSIM.

Introduction

The motivation for considering an LU block simulation method is discussed in detail in paper #304 in this report. To summarize: the simulation method of ore reserve calculation allows for the incorporation of secondary data, can be upscaled to any model resolution and can consider trends in the data. The major drawback of simulation techniques is the size of the necessary fine scale model and the requirement to store and process the realizations. The LU block simulation method proposed in this paper allows for the simultaneous generation of all necessary realizations and eliminates the need to store realizations.

BLUSIM is modified to incorporate simulation of multiple variables at multiple locations in one step. This will allow for the incorporation of (1) data of different types and (2) different variables. (1) Often there are discrepancies between blast hole and exploration drill hole data for the same variable. Incorporating a multivariate version of block LU simulation allows the modeler to consider data of different types simultaneously; for example, blasthole data would be used as a secondary correlated variable for the simulation of the more reliable diamond exploration data. (2) Often there are many secondary variables that can provide additional information to the variable of interest. In this instance the multivariate version of block LU simulation can be used to better inform the variable of interest with the secondary information.

A program, BLUSIM_MV, will be discussed that requires an LMC model to define the multivariate distribution between any number of data types and generates estimates of panel mean and variance with block LU simulation. The multiple variables used need not be colocated. A case study of a nickel deposit with four variables of interest will highlight the usage of the program in the multivariate sense.

Methodology

This methodology will consider a local independent LU simulation of the panels. Data near each panel are used to simulate in the panel and the corresponding panel mean and variance are stored. This eliminates the need to store the high resolution geostatistical model. Moreover, LU simulation is used because it is an efficient algorithm for a small number of nodes. It would be inappropriate to use LU simulation to generate a high resolution model of the entire field because of the size of the resulting fine scale model; therefore, each panel is discretized and LU simulation is performed independently within each panel to generate the fine scale model.

LU simulation has been developed for the joint simulation of multiple data locations (see paper 304 in this report), but it requires modification to consider simulating multiple data types at multiple locations within the panel. The LU simulation of multiple locations considering only one variable requires the following covariance matrix:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \qquad \textbf{(1)}$$

C contains four smaller sub matrices, C_{11}, C_{12}, C_{21} and C_{22} (with $C_{12}= C_{21}^T$). The C_{11} matrix contains the data to data covariance, equation 2. The C_{12} matrix contains the data to simulation location covariance, equation 3. The C_{22} matrix contains the simulation location to simulation location covariance, equation 4.

$$C_{11} = \begin{bmatrix} c(u_1,u_1) & \ldots & c(u_1,u_n) \\ \vdots & & \vdots \\ c(u_n,u_1) & \ldots & c(u_n,u_n) \end{bmatrix} \qquad \textbf{(2)}$$

$$C_{12} = \begin{bmatrix} c(u_1,v_1) & \ldots & c(u_1,v_m) \\ \vdots & & \vdots \\ c(u_n,v_1) & \ldots & c(u_n,v_m) \end{bmatrix} \qquad \textbf{(3)}$$

$$C_{22} = \begin{bmatrix} c(v_1,v_1) & \ldots & c(v_1,v_m) \\ \vdots & & \vdots \\ c(v_m,v_1) & \ldots & c(v_m,v_m) \end{bmatrix} \qquad \textbf{(4)}$$

where the *u* indicates the *n* data locations and *v* denotes the *m* simulation locations in the panel.

Now, consider the addition of multiple data types. The C_{11} matrix will now contain the cross-covariance between all data types and the number of simulation locations will increase. The matrices will have the following form:

$$C_{11} = \begin{bmatrix} c(u_1^1,u_1^1) & \ldots & c(u_1^1,u_{n_1}^1) & & c(u_1^1,u_1^p) & \ldots & c(u_1^1,u_{n_p}^p) \\ \vdots & & \vdots & \cdots & \vdots & & \vdots \\ c(u_{n_1}^1,u_1^1) & \ldots & c(u_{n_1}^1,u_{n_1}^1) & & c(u_{n_1}^1,u_1^p) & \ldots & c(u_{n_1}^1,u_{n_p}^p) \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ c(u_1^p,u_1^1) & \ldots & c(u_1^p,u_{n_1}^1) & & c(u_1^p,u_1^p) & \ldots & c(u_1^p,u_{n_p}^p) \\ \vdots & & \vdots & \cdots & \vdots & & \vdots \\ c(u_{n_p}^p,u_1^1) & \ldots & c(u_{n_p}^p,u_{n_1}^1) & & c(u_{n_p}^p,u_1^p) & \ldots & c(u_{n_p}^p,u_{n_p}^p) \end{bmatrix} \qquad \textbf{(5)}$$

$$C_{12} = \begin{bmatrix} c(u_1^1,v_1^1) & \ldots & c(u_1^1,v_{m_1}^1) & & c(u_1^1,v_1^p) & \ldots & c(u_1^1,v_{m_p}^p) \\ \vdots & & \vdots & \cdots & \vdots & & \vdots \\ c(u_{n_1}^1,v_1^1) & \ldots & c(u_{n_1}^1,v_{m_1}^1) & & c(u_{n_1}^1,v_1^p) & \ldots & c(u_{n_1}^1,v_{m_p}^p) \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ c(u_1^p,v_1^1) & \ldots & c(u_1^p,v_{m_1}^1) & & c(u_1^p,v_1^p) & \ldots & c(u_1^p,v_{m_p}^p) \\ \vdots & & \vdots & \cdots & \vdots & & \vdots \\ c(u_{n_p}^p,v_1^1) & \ldots & c(u_{n_p}^p,v_{m_1}^1) & & c(u_{n_p}^p,v_1^p) & \ldots & c(u_{n_p}^p,v_{m_p}^p) \end{bmatrix} \qquad \textbf{(6)}$$

$$C_{22} = \begin{bmatrix} c(v_1^1,v_1^1) & \ldots & c(v_1^1,v_{m_1}^1) & & c(v_1^1,v_1^p) & \ldots & c(v_1^1,v_{m_p}^p) \\ \vdots & & \vdots & \cdots & \vdots & & \vdots \\ c(v_{m_1}^1,v_1^1) & \ldots & c(v_{m_1}^1,v_{m_1}^1) & & c(v_{m_1}^1,v_1^p) & \ldots & c(v_{m_1}^1,v_{m_p}^p) \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ c(v_1^p,v_1^1) & \ldots & c(v_1^p,v_{m_1}^1) & & c(v_1^p,v_1^p) & \ldots & c(v_1^p,v_{m_p}^p) \\ \vdots & & \vdots & \cdots & \vdots & & \vdots \\ c(v_{m_p}^p,v_1^1) & \ldots & c(v_{m_p}^p,v_{m_1}^1) & & c(v_{m_p}^p,v_1^p) & \ldots & c(v_{m_p}^p,v_{m_p}^p) \end{bmatrix} \qquad \textbf{(7)}$$

where u^p indicates n_p data locations of variable type *p* and v^p indicates m_p simulation locations of variable type *p*.

Once these covariance matrices are set up traditional LU simulation can be used to decompose the resulting C matrix and generate multiple realizations at all locations for all *p* data types. Figure 1 more clearly demonstrates the meaning of the covariance matrices with an example. The symbols are used to denote data and simulation locations of different types and the necessary covariances can be visualized.

Implementation

BLUSIM_MV is a FORTRAN program that requires an input parameter file. The details of each line of the parameter file (Figure 2) follow:

Line 5 – Name of the data file.

Line 6 – The parameter file contains a separate line for EACH data point, even collocated data must be reformatted to be multiple entries in the data file. The variable type (ID) must also be indicated in the data file. The ID is 1 through to the number of variables *consecutively*, no ID numbers can be skipped. Only one column is required for the variable (ID is used to distinguish variable type). A declustering weight can also be added.

Line 7 – The number of variables must equal the highest ID of the variables in the data file. If fewer variables are selected to simulate, all variables are simulated and only the requested ones are output.

Line 8 – ID of the variables to simulate.

Line 9 – Trimming limits for the variables. The same trimming limits are placed on all data.

Line 10 – Option to transform the data. If the data are already transformed the output will average the normal scored values, it is better to let the program transform the data.

Line 11 – File for the transformation table to be output.

Line 12-14 – If desired, a transformation table can be included for the variables. This is in one file with each column representing a different variable.

Line 15-17 – Parameters to transform the variables. Each line must contain 2 parameters for each variable, for each of these lines there will be (nvar*2) parameters on the line.

Line 18-19 – Debugging information.

Line 20 – Output file name.

Line 21 – Number of realizations.

Line 22-24 – Grid defining the *panels*.

Line 25 – Number of discretization points per *panel*.

Line 26 – Number of selective mining units per panel, if number of discretization points divided by number of SMU's in each direction is not a whole number there will be fewer discretization points in some SMU blocks and the output will be nonsensical.

Line 27-30 – Cutoffs to report and option to consider equivalent grade. Requires a factor for each data *simulated*.

Line 31-32 – Option to write out all SMU realizations, if there are many realizations this file may be large.

Line 33 – Random number.

Line 34 – Minimum and maximum number of data to use in simulation. This can be different for each variable type considered. To consider only one min/max based on the total number of data rather than by data type, the third number should be set to -1.

Line 35 – Consider an octant search.

Line 36-37 – Search information. This will be used for all data types.

Remaining lines – These lines define an LMC. Any LMC defined must be positive definite and is checked by the program. To consider a Markov Model, specify it in the form of cross variograms (i.e. multiply the contributions of the structures by the correlation).

Case Study

The following case study will demonstrate the methodology. The available data is a nickel deposit with four variables (Ni, Fe, MgO and Co, data is shown in Figure 3). All variables will be simulated at the panel scale and the resulting distributions analyzed.

First, an LMC must be developed. It is often tedious to generate an LMC for four variables as a total of 10 variograms must be modeled while maintaining positive definiteness. VARFIT_LMC will be used to automatically fit the LMC of the 4 variables of interest. Figure 4 shows the resulting LMC after each variable has been normal score transformed.

The panel scale averages are shown in Figure 5. Moreover, BLUSIM (univariate) was considered for each variable where all variables are simulated independently; the resulting panel scale distributions can be compared to the multivariate case (Figure 6).

Recall that LU simulation was selected because it is efficient for solving small matrices. When considering multiple data types and large levels of discretization, the resulting LU simulation can become CPU intensive. Figure 7 shows how the CPU time for BLUSIM_MV varies with the input parameters. The basic parameters for the time trails follow:

- Number of discretization points per panel = 8 (2x2x2)
- Number of variables = 3
- Number of panels = 400
- Maximum number of data used = 15 (5+5+5)
- Number of realizations = 1000

For each graph in Figure 7 one of the above parameters is varied while the others remain constant.

The program CPU time is linearly dependent on the number of variables, panels and simulations; there is a cubic relationship to the number of data used in simulation and the number of discretization points. Examining the coefficients of the fitted functions in Figure 7 it can be seen that number of data types used has a large effect, an increase from 1 to 4 data types quintuples the CPU time. Moreover, the number of discretization points also has a significant impact on the CPU.

Conclusions and Future Work

A program to use LU simulation to simulate the panel scale distribution of multiple variables simultaneously has been presented. The panels are discretized and the mean, variance, grade above a cutoff and equivalent grade for a panel are calculated. This saves the memory requirements of storing the fine scale model. Each panel is simulated independently. The resulting panel distributions (Gaussian mean and variance) can be used in further transfer functions or to calculate reserves/resources.

Future work for this algorithm will focus on improving CPU speed. The first idea is to only simulate the variables of interest, currently all variables are simulated. The C_{22} matrix would consider the discretization points of only those variables that are to be output. This will reduce the size of the matrix if the number of variables of interest is smaller than the total number of variables available. The second idea is to invert a submatrix of C only once and will take advantage of the structure of the C matrix; C_{22} is constant. It would be possible to take the inverse of C_{22} once and use linear algebra with the inverse of C_{11} to find the inverse of C. Because C_{22} is a large portion of C, this will significantly improve CPU speed, at each panel only the inverse of the smaller matrix C_{11} is required.

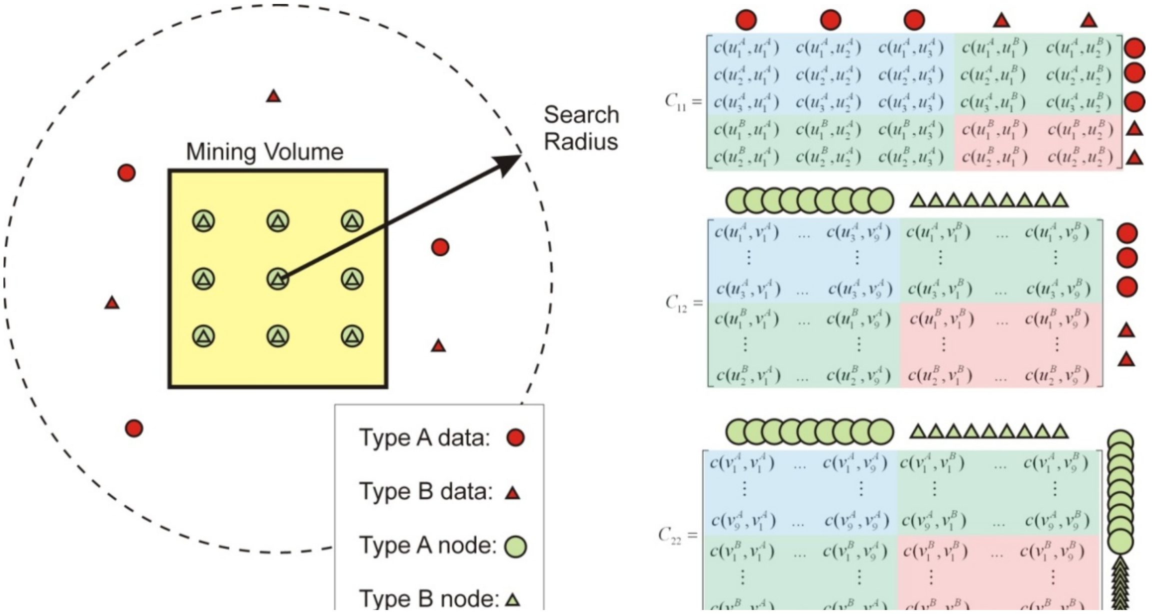


Figure 1: Example showing details of covariance matrices. The example on the left has 5 data (3 of type A and 2 of type B). Simulation will be performed in the panel (mining volume) at 18 simulation locations (9 of type A and 9 of type B). The resulting covariance matrices are shown on the right. Recall $C_{12}=C_{21}^{T}$.

```
                  Parameters for BLUSIM_MV
                  ************************

START OF PARAMETERS:
data.dat                  - file with data
4 5 6 9 8 10              - columns for X,Y,Z,varID, var,wt
3 2                       - number of variables, number to simulate
1  3                      - ID of variables to simulate
-1   1.0e21               - trimming limits
1                         - transform the data (0=no, 1=yes)
 blusim.trn               - file for output trans table
 0                        - consider ref. dist (0=no, 1=yes)
 histsmth.out             - file with ref. dist distribution
1  2 3 4 5 6 7 8          - columns for vr1 and wt1, vr2 and wt2...
0 1 1 9 2 20              - zmin,zmax(tail extrapolation)
1 20 1 20 1 20 1 20       - lower tail option, parameter
1 1 1 1 1 1 1 1           - upper tail option, parameter
0                         - debugging level: 0,1,2,3
blusim.dbg                - file for debugging output
blusim.out                - file for output
1000                      - number of realizations to generate
10  50 100                - nx,xmn,xsiz - Panel Size
10  50 100                - ny,ymn,ysiz
10  50  20                - nz,zmn,zsiz
3 3 3                     - nbx,nby,nbz- SMUs per Panel
6 6 6                     - x,y and z block discretization
2                         - number of cutoffs for reporting
4 5                       - cutoffs, will report statistics for information above
0                         - option to consider equivalent grade
1 0.8 0.6                 - for each variable SIMULATED need a factor
0                         - option to output all SMU realizations (0=no)
SMUreals.out              - file for the SMU realizations
67515                     - random number seed
1 10 1 10 1 10  1 10      - min, max data for Kriging
0                         - max per octant (0-> not used)
155 155 155               - maximum search radii
 0.0   0.0   0.0          - angles for search ellipsoid
1 1                       - Semivariogram for 1 and 1
2 0.001                   - nst, nugget effect
1 0.892 35.0 0.0 0.0      - it,cc,ang1,ang2,ang3
  37.796  30  3.7796      - a_hmax,a_hmin,a_vert
1 0.107 35.0 0.0 0.0      - it,cc,ang1,ang2,ang3
```

```
43   195.610  156.488  19.56 - a_hmax,a_hmin,a_vert
44 1 2                       - Semivariogram for 1 and 2
45 2 0.001                   - nst, nugget effect
46 1 0.340 35.0 0.0 0.0      - it,cc,ang1,ang2,ang3
47   37.796   30  3.7796     - a_hmax,a_hmin,a_vert
48 1 0.239 35.0 0.0 0.0      - it,cc,ang1,ang2,ang3
49   195.610   156.488 19.56 - hmax,a_hmin,a_vert
50 1 3                       - Semivariogram for 1 and 3
51 2 0.001                   - nst, nugget effect
52 1 -0.295 35.0 0.0 0.0     - it,cc,ang1,ang2,ang3
53   37.796    30  3.7796    - a_hmax,a_hmin,a_vert
54 1 -0.170 35.0 0.0 0.0     - it,cc,ang1,ang2,ang3
55   195.610   156.488 19.56 - hmax,a_hmin,a_vert
56 2 2                       - Semivariogram for 2 and 2
57 2 0.001                   - nst, nugget effect
58 1 0.426 35.0 0.0 0.0      - it,cc,ang1,ang2,ang3
59   37.796    30  3.7796    - hmax,a_hmin,a_vert
60 1 0.573 35.0 0.0 0.0      - it,cc,ang1,ang2,ang3
61   195.610   156.488 19.56 - hmax,a_hmin,a_vert
62 2 3                       - Semivariogram for 2 and 3
63 2 0.001                   - nst, nugget effect
64 1 -0.409 35.0 0.0 0.0     - it,cc,ang1,ang2,ang3
65   37.796    30  3.7796    - hmax,a_hmin,a_vert
66 1 -0.391 35.0 0.0 0.0     - it,cc,ang1,ang2,ang3
67   195.610   156.488 19.56 - hmax,a_hmin,a_vert
68 3 3                       - Semivariogram for 3 and 3
69 2 0.001                   - nst, nugget effect
70 1 0.675 35.0 0.0 0.0      - it,cc,ang1,ang2,ang3
71   37.796    30  3.7796    - hmax,a_hmin,a_vert
72 1 0.324 35.0 0.0 0.0      - it,cc,ang1,ang2,ang3
73   195.610   156.488 19.56 - hmax,a_hmin,a_vert
```

Figure 2: Parameter file for BLUSIM_MV.

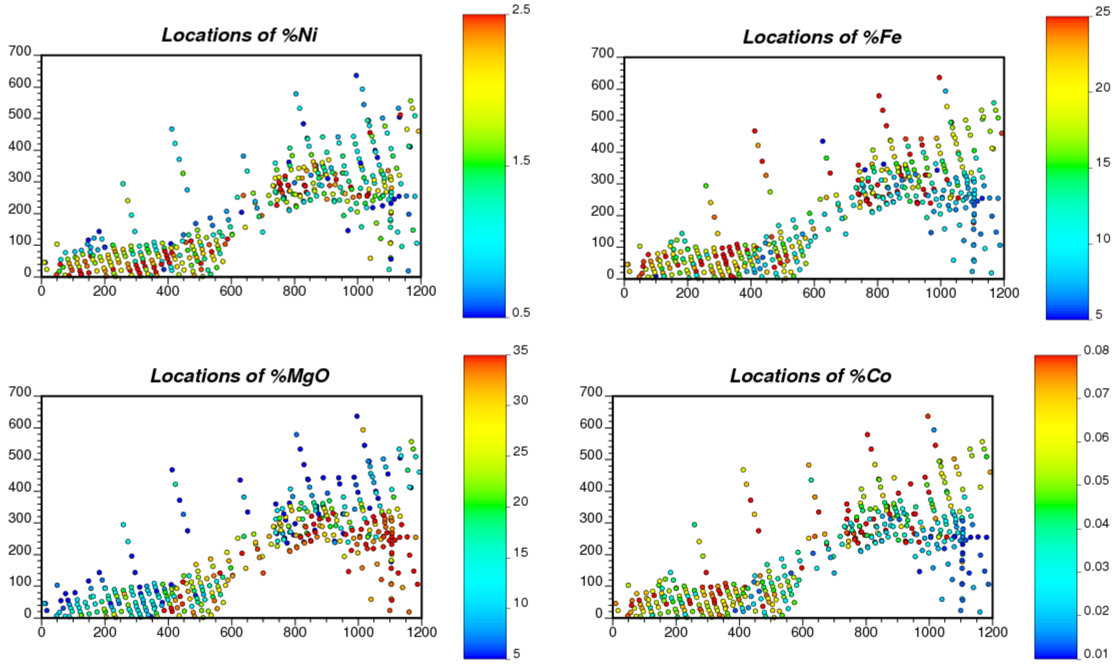
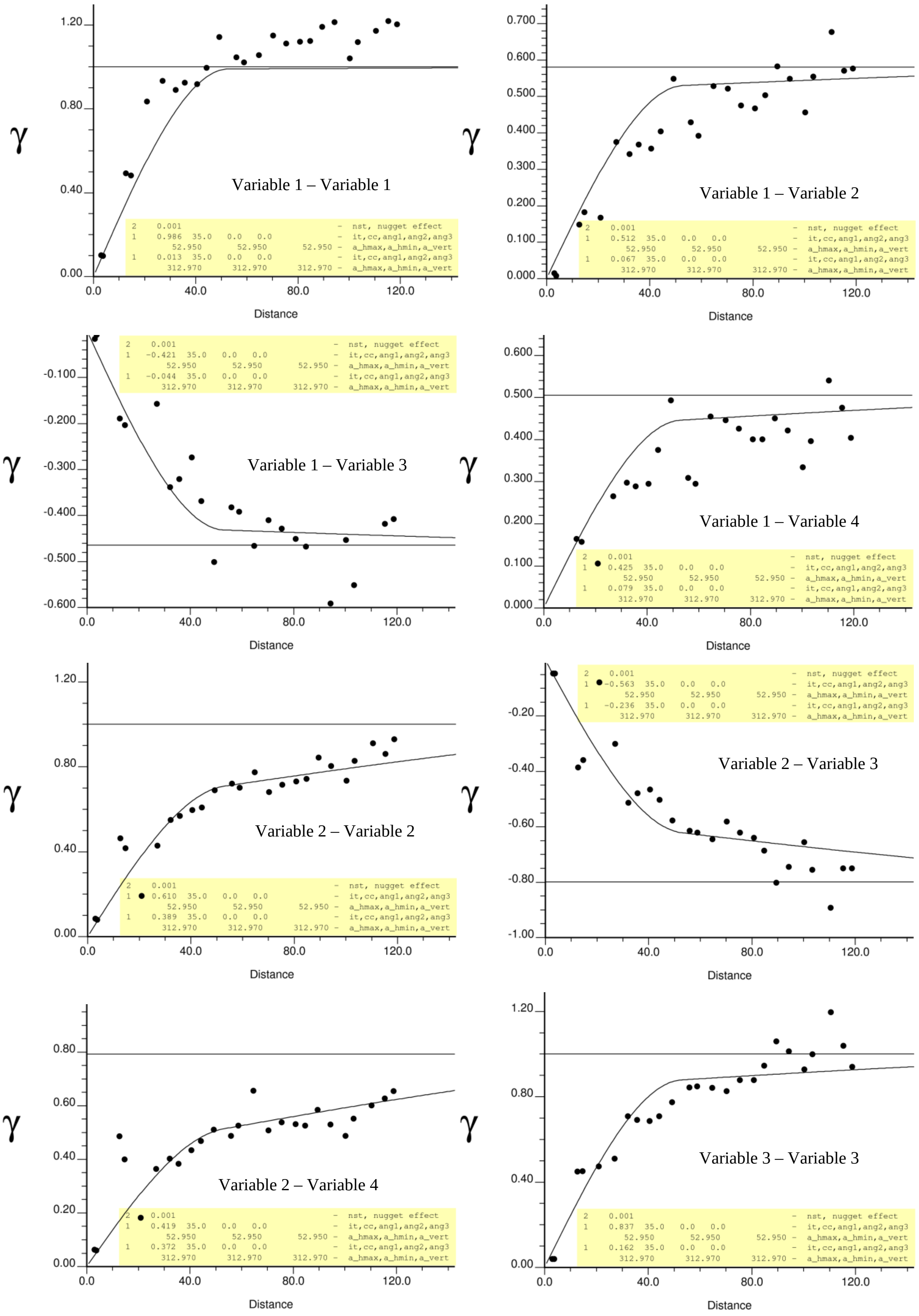


Figure 3: Location of data for the case study.



```
2    0.001                                      -  nst, nugget effect
1    0.986  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
       52.950        52.950        52.950 -  a_hmax,a_hmin,a_vert
1    0.013  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
      312.970       312.970       312.970 -  a_hmax,a_hmin,a_vert
```

```
2    0.001                                      -  nst, nugget effect
1    0.512  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
       52.950        52.950        52.950 -  a_hmax,a_hmin,a_vert
1    0.067  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
      312.970       312.970       312.970 -  a_hmax,a_hmin,a_vert
```

```
2    0.001                                      -  nst, nugget effect
1   -0.421  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
       52.950        52.950        52.950 -  a_hmax,a_hmin,a_vert
1   -0.044  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
      312.970       312.970       312.970 -  a_hmax,a_hmin,a_vert
```

```
2    0.001                                      -  nst, nugget effect
1    0.425  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
       52.950        52.950        52.950 -  a_hmax,a_hmin,a_vert
1    0.079  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
      312.970       312.970       312.970 -  a_hmax,a_hmin,a_vert
```

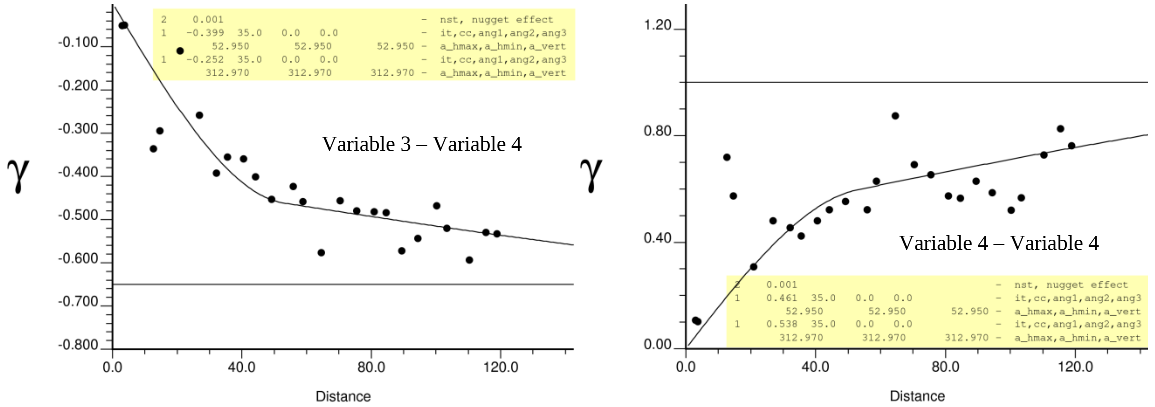
```
2    0.001                                      -  nst, nugget effect
1    0.610  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
       52.950        52.950        52.950 -  a_hmax,a_hmin,a_vert
1    0.389  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
      312.970       312.970       312.970 -  a_hmax,a_hmin,a_vert
```

```
2    0.001                                      -  nst, nugget effect
1   -0.563  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
       52.950        52.950        52.950 -  a_hmax,a_hmin,a_vert
1   -0.236  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
      312.970       312.970       312.970 -  a_hmax,a_hmin,a_vert
```

```
2    0.001                                      -  nst, nugget effect
1    0.419  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
       52.950        52.950        52.950 -  a_hmax,a_hmin,a_vert
1    0.372  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
      312.970       312.970       312.970 -  a_hmax,a_hmin,a_vert
```

```
2    0.001                                      -  nst, nugget effect
1    0.837  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
       52.950        52.950        52.950 -  a_hmax,a_hmin,a_vert
1    0.162  35.0   0.0   0.0                    -  it,cc,ang1,ang2,ang3
      312.970       312.970       312.970 -  a_hmax,a_hmin,a_vert
```

Figure 4: 10 variograms and cross variograms that define the LMC for 4 variables of interest (variable 1 = Ni, variable 2 = Fe, variable 3 = MgO, variable 4 = Co).



```
2    0.001                                    -  nst, nugget effect
1   -0.399  35.0   0.0   0.0                  -  it,cc,ang1,ang2,ang3
       52.950       52.950       52.950 -  a_hmax,a_hmin,a_vert
1   -0.252  35.0   0.0   0.0                  -  it,cc,ang1,ang2,ang3
      312.970      312.970      312.970 -  a_hmax,a_hmin,a_vert
```

```
2    0.001                                    -  nst, nugget effect
1    0.461  35.0   0.0   0.0                  -  it,cc,ang1,ang2,ang3
       52.950       52.950       52.950 -  a_hmax,a_hmin,a_vert
1    0.538  35.0   0.0   0.0                  -  it,cc,ang1,ang2,ang3
      312.970      312.970      312.970 -  a_hmax,a_hmin,a_vert
```

Figure 4 continued: 10 variograms and cross variograms that define the LMC for the 4 variables of interest (variable 1 = Ni, variable 2 = Fe, variable 3 = MgO, variable 4 = Co).

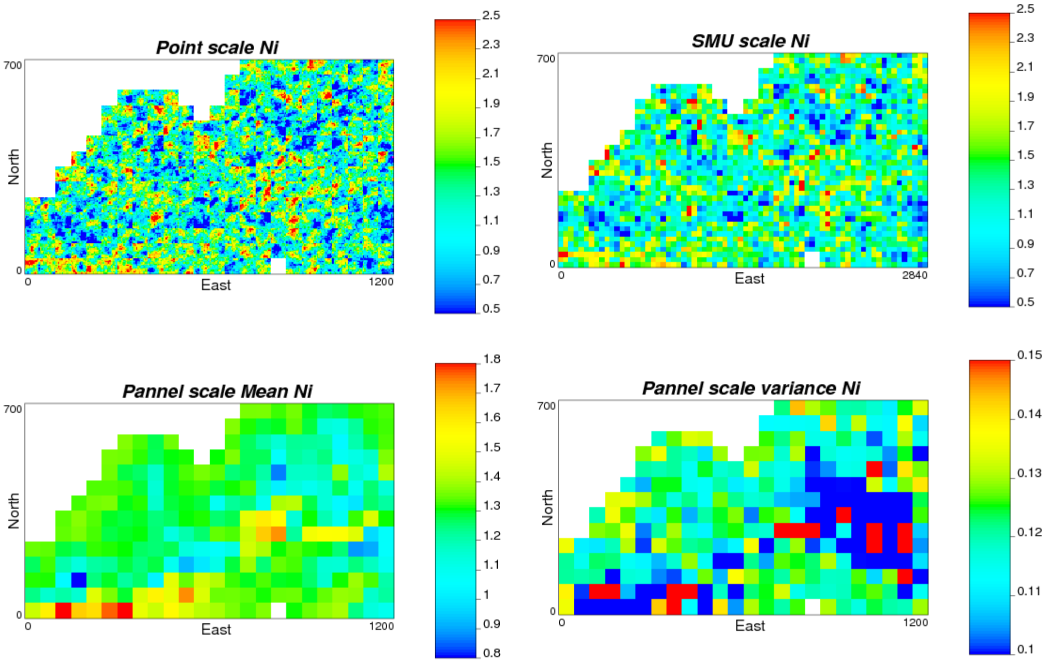


Figure 5: Above left, one point scale realization of Nickel. Above right, one SMU scale realization of Nickel. Below left, the panel scale mean of Nickel, each panel is the average of 1000 realizations at the point scale. Below right, the panel scale variance of Nickel.

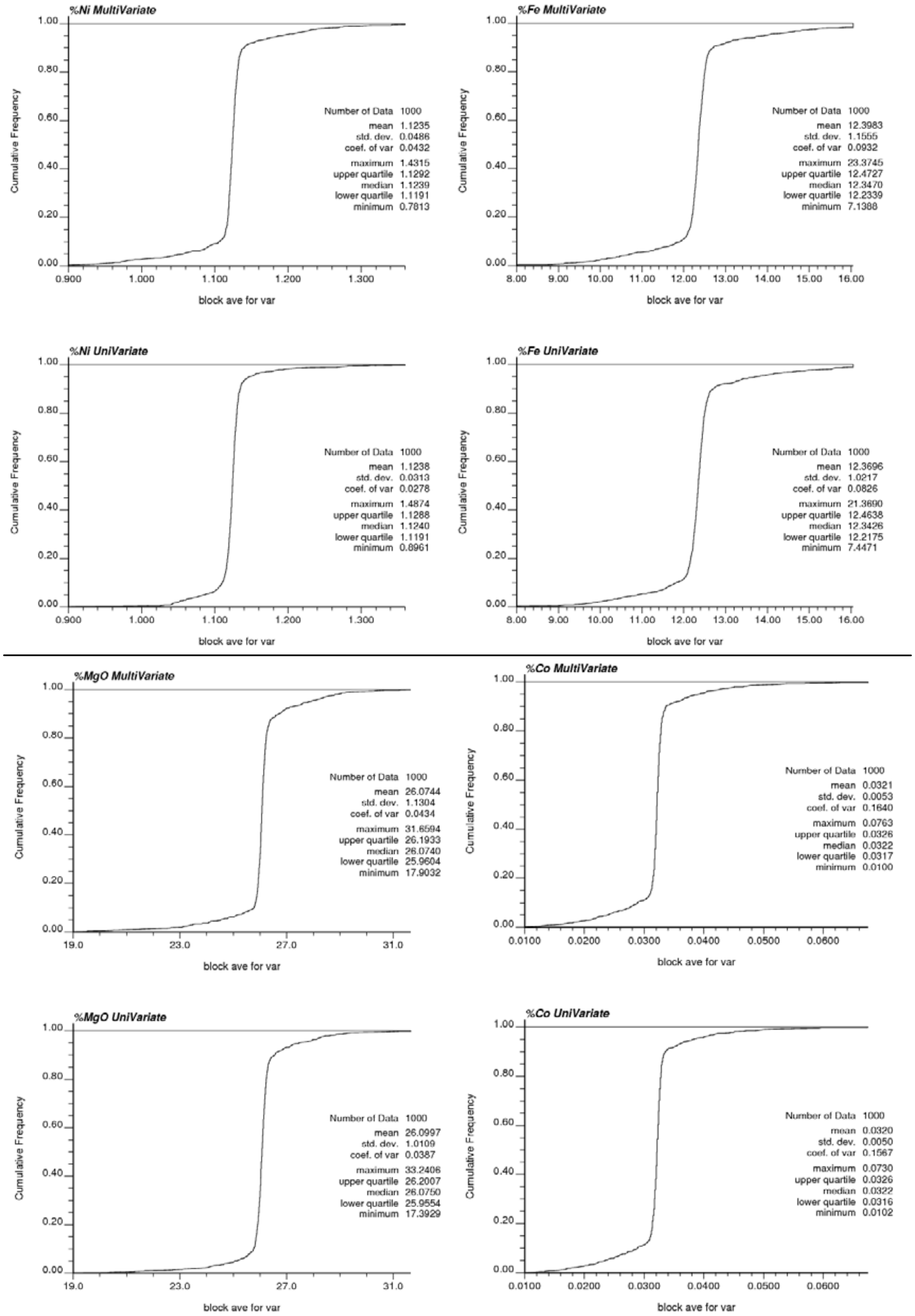


Figure 6: Above, panel scale distributions considering the LMC modeled (BLUSIM_MV). Below, panel scale distributions considering that each variable is modeled independently (BLUSIM).

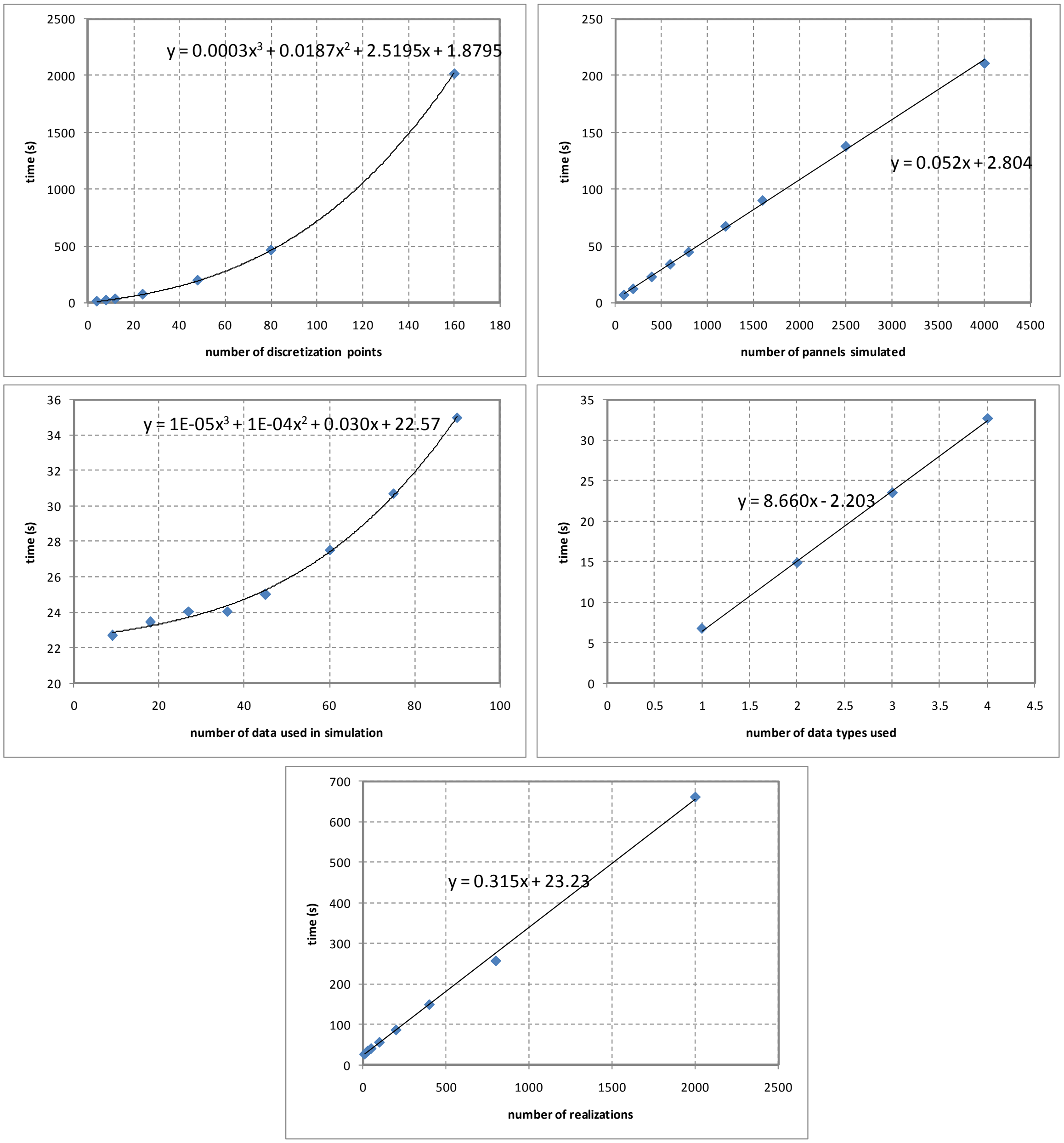


Figure 7: Time trials for the case study. The effect of the number of discretization points, panels, data, data types used, and realizations is on the CPU time is shown.