# Progress Towards Fast Block LU Simulation on Large Grids

Tong Wang

*The Lower –Upper Simulation is an important conditional simulation method for several correlated variables. It is efficient for large grids if applied on a block-by-block basis. For multivariable, a model of coregionalization is needed to do the simulation. Although cross-covariances are the key for integration secondary data, modeling of cross-covariance by fitting the linear model of coregionalization (LMC) is considered a cumbersome task, which is the reason that the LMC is limitedly used in practice. An approximate model of coregionalization is presented for doing LU with multiple variables. This short note is the start of a Masters level research project.*

## Introduction

Simulated realizations are useful to obtain a realistic picture of spatial variability. There are many methods for modeling multiple variables. The major drawback is that they are all very computational expensive for large grids. If the simulation is done at a block-support, the speed could be raised a lot. The Block LU simulation is an alternative. Blocks are considered independently and the computation of recoverable reserves is done at an SMU scale. (see references). This approach locally simulates the grade values via the matrix decomposition method (Davis, 1987).

There could be many variables available in the domain of interest. The block simulation with multiple variables is possible, but requires a positive definite linear model of coregionalization. A new methodology is proposed to use the n direct variograms and the correlation matrix between the n variables. The covariance between different variables may need to be corrected. A test case is developed.

## Methodology

The LU method for one variable is as follows:

- Transform the data to a standard Gaussian distribution
- Calculate and model the variogram
- Compute covariance matrix C for the location in the search neighborhood
- Decompose the covariance via the Cholesky decomposition into a lower and upper triangular matrix, where the upper triangular matrix is the transpose of the lower triangular matrix $C = L\mathrm{U}$ where $\mathrm{L'=U}$
- Now consider a random vector $\mathrm{y = Lw}$ where w is a vector of independent N (0, 1) distributed random numbers
- Back-transform the simulated values to original units.

Multivariate block LU simulation is obtained by using the LMC. The method is developed for the joint simulation of multiple data locations. For simplification, say 3 variables, v, u and w. The covariance matrix of data location the grid to be generated and the partition matrix as following:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Where $C_{11}$ is the covariance and cross-covariance between data points, $C_{12}$ is covariance and cross-covariance between search data points and grid discretization points or grid points. The $C_{22}$ is the covariance and cross-covariance matrix between discretization location. Each sub matrix in $C_{11}, C_{12}, C_{22}$, contains the covariance among the n v, u, and w data and cross covariance between uv, uw and vw.

$$C_{11}=\begin{bmatrix} c_{u1,u1} & \cdots & c_{u1,up} & c_{u1,v1} & \cdots & c_{u1,vp} & c_{u1,w1} & \cdots & c_{u1,wp} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{up,u1} & \cdots & c_{up,up} & c_{up,v1} & \cdots & c_{up,vp} & c_{up,w1} & \cdots & c_{up,wp} \\ & & & c_{v1,v1} & \cdots & c_{v1,vp} & c_{v1,w1} & \cdots & c_{v1,wp} \\ & & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & & c_{vp,v1} & \cdots & c_{vp,vp} & c_{vp,w1} & \cdots & c_{vp,wp} \\ & & & & & & c_{w1,w1} & \cdots & c_{w1,wp} \\ & & & & & & \vdots & \ddots & \vdots \\ & & & & & & c_{wp,w1} & \cdots & c_{wp,wp} \end{bmatrix}$$

$$C_{12}=\begin{bmatrix} c_{u1,u1} & \cdots & c_{u1,um} & c_{u1,v1} & \cdots & c_{u1,vm} & c_{u1,w1} & \cdots & c_{u1,wm} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{up,u1} & \cdots & c_{up,um} & c_{up,v1} & \cdots & c_{up,vm} & c_{up,w1} & \cdots & c_{up,wm} \\ & & & c_{v1,v1} & \cdots & c_{v1,vm} & c_{v1,w1} & \cdots & c_{v1,wp} \\ & & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & & c_{vp,v1} & \cdots & c_{vp,vm} & c_{vp,w1} & \cdots & c_{vp,wm} \\ & & & & & & c_{w1,w1} & \cdots & c_{w1,wm} \\ & & & & & & \vdots & \ddots & \vdots \\ & & & & & & c_{wp,w1} & \cdots & c_{wp,wm} \end{bmatrix}$$

$$C_{22}=\begin{bmatrix} c_{u1,u1} & \cdots & c_{u1,um} & c_{u1,v1} & \cdots & c_{u1,vm} & c_{u1,w1} & \cdots & c_{u1,wm} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{um,u1} & \cdots & c_{um,um} & c_{um,v1} & \cdots & c_{um,vm} & c_{u1,w1} & \cdots & c_{um,wm} \\ & & & c_{v1,v1} & \cdots & c_{v1,vm} & c_{v1,w1} & \cdots & c_{v1,wm} \\ & & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & & c_{vm,v1} & \cdots & c_{vm,vm} & c_{vm,w1} & \cdots & c_{vm,wm} \\ & & & & & & c_{w1,w1} & \cdots & c_{w1,wm} \\ & & & & & & \vdots & \ddots & \vdots \\ & & & & & & c_{wm,w1} & \cdots & c_{wm,wm} \end{bmatrix}$$

Where suppose there are p data in the search radius and m data discretization in the grid. The above covariance among the 3 variables is retreived from their cross variogram model.

The new algorithm needs the new cross variogram model between the variables. It is obtained by multiply correlation coefficient and the average of the two direct variogram.

$$\gamma_{k,k'}=\rho_{k,k'}\bullet\frac{\gamma_k+\gamma_{k'}}{2}$$

But the matrix for LU is not always positive definite, that is the variance must never be negative. The covariance function must insure the previous variance will always be none negative. Such we see that the covariance is positive definite. (Journel and Huijbregts, 1978 ) We come up with a method that can fix the matrix to ensure the big matrix is always positive definite.

According to the theorem, a symmetric matrix, say C, is positive definite if and only if all the eigenvalue of C are positive. So, when correcting this matrix:

1. The Krylov subspace methods are used to calculate the approximate minimum eigenvalue $\lambda_n$. The reason why Krylov subspace methods is used is that it is when the matrix in question is real and symmetric the lanczos algorithm provides an efficient method for producing an orthonormal sequence of vectors which provides a basis for an appropriate Krylov subspace (Chris C. Paige 1995).
2. If the minimum eigenvalue is negative, the symmetric matrix C is not positive definite. We, then, correcting the matrix by adding $2\lambda_n$ and a small number $\varepsilon = 0.02$ to the diagonal.
3. Move to the next node, and build another new big matrix, calculate the approximate minimum eigenvalue, check, if it is not positive definite, correct the diagonal element in the diagonal to make sure it is positive definite.

**Test Case**

A synthetic 2D data is considered to test this method (Figure 1). In this case, there are 18 data in the search neighborhood, and the fairly large gird has a discretization 16 by 16 as showing. In the search neighborhood, we assume there are there 3 variables u, v and w. and they have a specific correlation matrix. They have three variogram models for this 2D case. For u variable it has the variogram model $\gamma_u = Sph(h)_{a=100}$.

For v variable it has the variogram model:

$$\gamma_v = Exp(h)_{a=100}$$

For w variable it has the variogram model:

$$\gamma_w = Sph(h)_{a=125}$$

The cross-variogram of uv is:

$$\gamma_{uv} = \rho_{uv} \bullet \frac{\gamma_u + \gamma_v}{2}$$

The cross-variogram of uw is:

$$\gamma_{uw} = \rho_{uw} \bullet \frac{\gamma_u + \gamma_w}{2}$$

The cross-variogram of vw is:

$$\gamma_{vw} = \rho_{vw} \bullet \frac{\gamma_v + \gamma_w}{2}$$

The big covariance matrix is built using the method of block LU simulation showing above. This big matrix is 102 by 102. After building the matrix the minimum eigenvalue are calculated, which is -1.049, then we adding two times of the minimum eigenvalue -1.049 plus the e=0.02 to the diagonal elements of the matrix. After calculation the minimum eigenvalue is 0.00107. So we make sure that the matrix is positive. From this simple test, we want to known the time consumption if we do the LU decomposition 1000 times. The time is also captured, that is 7644 seconds.

**Conclusion and Future Work**

A new algorithm is proposed by using the k direct variogram and the correlation matrix. The minimum approximate eigenvalue is calculated by the Krylov subspace algorithm and the correction matrix is made by adding two times of the approximate eigenvalue and a small number. The future work is to focus on the program to implement of this algorithm. Also we notice that the C22 matrix in each time is almost the same, how to iterative update this to next grid could also improve the CPU time.

## References

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

Julian Ortiz. and Clayton V. Deutsch. (2007) Local Recoverable Reserves Prediction with Block LU simulation. CCG report 9. University of Alberta.

Jeff Boisvert and Clayton V. Deutsch. (2007) Multivariate Block LU simulation CCG report 9. University of Alberta.

Chris C. Paige. Approximate Solutions and Eigenvalue Bounds from Krylov Subspaces. Numerical Linear Algebra with Applications, Vol. 2(2), 115-133 (1995)

Michael W. Davis Production of Conditional Simulations via the LU Triangular Decomposition of the Covariance Matrix Mathematical Geology, Vol. 19, No. 2, 1987

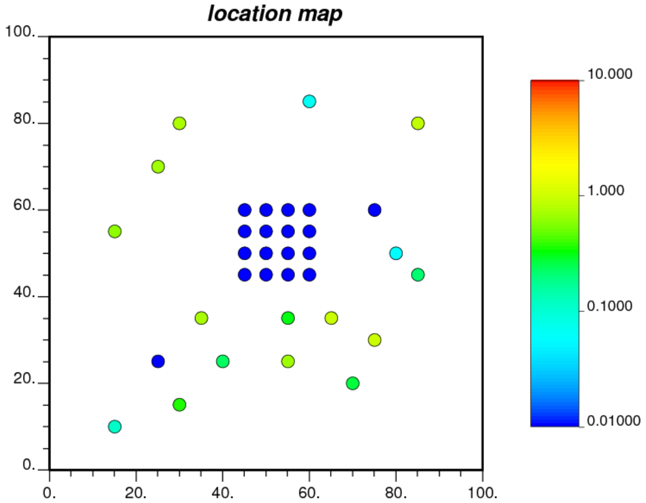

**Figure 1**: The location map of the synthetic data.