Using Cholesky Decomposition and Sparse Matrices for Conditional Simulation of a Gaussian 2D Random Field

Dr.Eng. Mohammad Saleh Al-Abdallah
Faculty of Civil Engineering
Damascus University

Abstract

This study presents an efficient practical method for the generation of sequential conditional simulation of a Gaussian two-dimensional random field which we frequently encounter in GIS spatial analysis problems such as DEM's generation from a limited number of data. The many realizations typically correspond to many reasons such as the geospatial uncertainty, the morphological perturbations over the surface having a complex structure or the inadequate representation of the triangulated network TIN or grid. These realizations with simulation-based concept enable the performance and uncertainty assessment that tunes to various geospatial (GIS) applications. For DEM generation and implementation of the conditional simulation, we need to decompose the c*ovariance matrix* of the data points and grid nodes by *Cholesky Decomposition*. Conditional simulation respect data values and transfers those values into the grid nodes. With the Incomplete Cholesky decomposition of the covariance matrix, we can produces as many simulations as needed in a single step with an accuracy, in a global sense, much better than the *Moving Window Kriging* method. In other words, we don't need to repeat covariance matrix generation and decomposition many times. On the other hand, there is the problem of producing covariance matrices in the case of large dataset, which proved to be time consuming and may take several hours on PC. The present paper presents a solution to this problem using *Sparse Matrices Technique* and *Cholesky decomposition* to achieve conditional simulation, reducing the time required for computations dramatically, as well as decreasing the demand of large amount of computer memory. For the purpose of this study and testing all algorithms, a *MATLAB* Programs was made by the author. They have been used in all computation stages and applied using real data. The study has shown that we can reduce computation time by 85%-95% according to the scale of the problem yet saving a considerable space in memory needed to store matrices.

استخدام تحليل تشولسكي و المصفوفات شبه الفارغة لإجراء محاكاة مشروطة لحقل عشوائي ثنائي يتبع غاوص

د. م محمد صالح العبدالله

كلية الهندسة المدنية – جامعة دمشق

ملخص البحث

يقدم هذه البحث طريقة عملية فعالة من أجل توليد سيناريوهات محاكاة مشروطة متتابعة لحقل ثنائي الأبعاد بمتحول عشوائي يتبع غاوص وهو ما نقابله بصورة متكررة عند إجراء التحليل المكاني في أنظمة المعلومات الجغرافية GIS. كمثال توليد النموذج الإرتفاعي العددي DEM لسطح الأرض إنطلاقاً من عدد محدود من نقاط الرصد. نلجأ لهذا النوع من المحاكاة لعدة أسباب: عدم الموثوقية الجيومكانية، البنية المورفولوجية المضطربة أو المعقدة للسطح، استخدام تمثيل شبكي مثلثاتي TIN أو شبكي تربيعي (grid) غير دقيق ..الخ. نستطيع من خلال هذه السيناريوهات التي تعتمد مفهوم المحاكاة تقدير الأداء و الموثوقية في التمثيل الذي يتناغم مع العديد من التطبيقات الجيومكانية في GIS. لإجراء المحاكاة الشرطية للنموذج الإرتفاعي العددي DEM نحتاج إلى تحليل مصفوفة التباين بطريقة تشولسكي لكل من البيانات ونقاط الشبكة التربيعية. تحترم المحاكاة الشرطية قيم البيانات وتقوم بتحويل هذه القيم إلى النقاط الجديدة للشبكة (grid). يساعدنا تحليل تشولسكي غيرالتام في توليد سيناريوهات محاكاة عديدة وذلك بخطوة واحدة و بدقة تفوق الدقة الناجمة عن كريجينغ بالنافذة المتحركة. هذا يعني أننا لا نحتاج لإعادة تشكيل وتحليل مصفوفة التباين مرات عديدة. هنا تبرز مشكلة حساب مصفوفة التباين الذي يمكن أن يأخذ وقتاً طويلاً (عدة ساعات) عند التعامل مع بيانات شبكة بحجم كبير نسبياً. يُقدم هذا البحث حلاً لهذه المشكلة عن طريق استخدام تقنية المصفوفات شبه الفارغة وتحليل تشولسكي لإنجاز المحاكاة الشرطية بصورة يتم فيها تقليل زمن الحساب بصورة جذرية كما أنها نقلل من الحاجة إلى حيز كبير من ذاكرة الحاسب. بقصد اختبار هذه المنهجية الجديدة تم وضع برامج بلغة ماتلاب من قبل الباحث أنجزت كافة مراحل الحساب بعد تطبيقها على بيانات حقلية فعلية. أظهرت هذه الدراسة أنه يمكننا إختزال من 85% إلى 95% من زمن الحساب حسب حجم المشكلة، إضافة إلى التوفير الكبير في حجوم التخزين للمصفوفات.

Key Words: *Sparse Matrices, Incomplete Cholesky Decomposition, Geostatistical Simulation, Gaussian Random Fields, Spatial Data Analysis in GIS, DEM.*

Introduction

Recent advances in geographical information systems and global positioning systems enable accurate geocoding of locations where scientific data are collected. This has encouraged the formation of considerable amount of data sets in many fields and has generated considerable interest in statistical modeling for location-referenced spatial data [Chiles J., Delfiner P. (1999) Møller (2003), Banerjee et al. (2004) and Schabenberger and Gotway (2004)] for a variety of methods and applications. *Geostatistics* is a subset of statistical method specialized in analysis, interpretation of geographically referenced data (Goovaerts, P. 1997). Cressie (1993) considers geostatistics to be only one of the three scientific fields specialized in the analysis of spatial data. In the most pragmatic terms, geostatistics is an analytical tool for statistical analysis of sampled field data (Bolstad, 2007). Today, geostatistics is not only used to analyze point data, but also increasingly in combination with various GIS data sources: e.g. to explore spatial variation in remotely sensed data, to quantify noise in images and for their enhancement and filtering (e.g. filling of the missing pixels), to improve *Digital Elevation Models* (DEM), ***generation or simulation DEM's***, optimize spatial sampling (Brus and Heuvelink, 2007), selection of best spatial resolution for image data and selection of support size for ground data [Atkinson and Quattrochi, et al (2000)]. Many application fields use geostatistics for spatial data analysis and interpretation like: (1) geosciences, (2) water resources, (3) environmental sciences, (4) agriculture, forestry (5) soil science, (6) mathematics & statistics, (7) ecology, (8) civil and petroleum engineering (10) meteorology [Hengl T. (2007)] ,[Lantuejoul C. (2002)],[Mund Jan-Peter (2013)].

Full inference and accurate assessment of uncertainty often require Markov chain Monte Carlo (MCMC) methods [David P. Landau, Kurt Binder (2009), Banerjee et al., (2008)]. However, such fitting involves matrix decompositions whose complexity increases as $O(n^3)$ (n is the number of locations) at every iteration of the MCMC algorithm; hence the infeasibility or '*big n*' problem for large data sets. Evidently, the problem is further aggravated when we have a vector of random effects at each location. Spatial process models for analyzing geostatistical data entails computations that become prohibitive as the number of spatial locations become large. In addition geostatistical modeling usually involves many variables and many locations.

The suggested LU simulation method for generating realizations (or DEM simulations), involves producing covariance matrices that are too large and not necessarily amenable to direct decomposition, inversion or manipulation. This paper presents an efficient Implementation method that uses LU decomposition or Cholesky Method for symmetric matrices, as well as ***Sparse Matrix Techniques*** for generating conditional realizations using randomized methods. ***Sparse Matrix Technique*** can overcome the problem of covariance matrices of huge sizes. This technique reduces the time required for large-scale systems computations including the Eigenvalue problem as well as the demand of large amount of memory. The LU method has some other advantages over other methods, such as the Turning Bands (TB) or FFT, in that the simulation and conditioning are implemented simultaneously [Gneiting Tilmann et al. (2005)]. In addition LU algorithm is considered much more simple, fast and easy to implement [Dietrich, C. R. (1993)]. LU decomposition Method due to its author Davis (1986), assumes that all grid nodes will be simulated at the same time and that all available data will be used. Using Sparse Matrix Techniques with an approximate incomplete decomposition method in the simulation of larger grid schemes, or large covariance matrices, is the only way out to overcome the computational machine errors [Dietrich, C. R. and Newsam, G. N. (1997).]. In general, the covariance matrix of order 1000 or more is considered as sparse [Cressie, N. and Huang, H.-C. (1999)] The purpose of this paper is to implement Conditional Simulation by LU decomposition (or Cholesky decomposition method) in combination with Sparse Matrices Technique to generates realization of N random variables at n spatial locations usually from a grid structure, using Monte Carlo Method (MCMC) and preserving the data values at original locations within the predefined spatial structure. On the other hand, *Sequential Simulation* algorithms are considered the most frequently used techniques, having several advantages over other methods, including the automatic handling of anisotropy as well as data conditioning. Their theoretical basis is simple and it can be applied to many simulations problems with single variable as well as with multiple variables, either continuous or categorical.

Conditional Simulation Concept

The building blocks of a conditional simulation are the mean function $\mu(\cdot)$, the covariance function $\mathrm{C}(\cdot)$ and the most important data vector z_d. It is required that the conditionally simulated process $Z_{sc}(x)$ pass through the data z_d, having unconditional mean $\mu(\cdot)$ and variance $\mathrm{C}(\cdot)$. One might think that the *kriging* predicator $Z_{ak}(x)$ would satisfy the requirements, because it does interpolate the data exactly and it is unbiased. However as *kriging* has a smoothing tendency, it does not possess enough variability in order to give a posterior probability distribution about the uncertainty [Myers, J .C., (1997)].

The purpose of *Conditional Simulation* is to produce *Random Fields* that simulate the spatial variability of the underlying random process $Z(x)$ [Malinowski A, Schlather M, Menck PJ (2015)],[Cressie, N.; Wikle, C.K. (2011)], [Isaak, E.H. & Srivastava RM. (1989)].

Theoretically, with *Conditional Simulation* we are able to generate an infinite number of possible realizations of a *Random Field* $\{ Z_s(x), x \epsilon D, s = 1 \rightarrow \infty \}$. From among the infinite simulations we choose those that meet certain condition $Z_s(x_a) = Z_0(x_a), \forall x_a \in D$. For example if we want to simulated a model that honors data values at the actual data locations, we set:

$Z_{cs}(x_a) = Z_0(x_a), \forall x_a \in D$, Where x_a represents data locations.

This is known as *Conditional Simulation*, which has the same variability characteristics as the real observed phenomenon. This means that the simulated values $Z_{cs}(x_a)$ have the same first two experimentally found moments (namely the mean and the variance or *Variogram*) as the real values $Z_0(x_a)$. On the other hand, if not then the simulated values $Z_{cs}(x_a)$ are not the best possible estimators of the random process $Z(x)$. Journel and Huijbregts (1978) showed that the posterior estimation variance of *Conditional Simulation* is as twice as that of *Kriging*, thus one should emphasize that the objective of simulation is not to obtain the best *unbiased estimator* provided by *Kriging*. *Conditional Simulation* is useful to get some idea of the amount of variability remaining in the physical model or process $Z(x)$ conditioning with respect to the observations [Journel, A.G (1989)]. Thus predictions and simulations address two different problems.

Now consider the decomposition of the process into a *kriging predicator* and *unconditional residual* [Journel, A.G. & Huijbregts, C. (1978)].

$$Z_{cs}(x) = Z^*(x) + [Z_{us}(x) - Z^*_{us}(x)] \quad (1)$$

Where $Z_{cs}(x)$ is the conditional simulation, $Z^*(x)$ is the kriging estimators using the real data set (representing the estimated grid), $Z_{us}(x)$ is the unconditional simulation, and $Z_{sk}(x)$ is simple-kriging estimators using the unconditional simulated data. The two components of the right-hand side of $Z^*(x)$ and $Z_{us}(x) - Z^*_{us}(x)$ are orthogonal. This orthogonality implies that $Z_{cs}(x)$ has the same unconditional covariance as (x), namely$C(\cdot)$. The quantity $Z_{us}(x) - Z^*_{us}(x)$ can be obtained by kriging the difference between data values and the unconditionally simulated ones at data locations. Thus the above expression can rewritten as follows [Davis (1987) and Cressie (1993)]

$$Z_{cs}(x) = Z_{us}(x) + c(x)' \cdot \sum^{-1} (z_d - z_{us}) \quad (2)$$

Where $Z_{cs}(x)$ and $Z_{us}(x)$ has the same meaning given above,

$c(x)' \equiv C(x_d, x_g), \forall x_d \in D, \forall x_g \in G$: is the covariance vector between data nodes and grid nodes.

$\sum \equiv Var(z)$: is the variance-covariance matrix between the data and itself. z_d *and* z_g are two vectors representing actual data and the simulated ones at the data node locations.

Conditional Sequential Simulation

The principle of *Conditional Sequential Simulation* is once the new value simulated, it is added to the original set of conditioning data, and the procedure repeated [Gomez-Hernandez, J.J , Cassiraga E.F. (1994)]. Finally all simulated nodes (by construction) will have the same initial spatial structure provided that all node values at data locations preserved. The principle of *Sequential Simulations* can be described as follows [Christakos, G. (2005)]: Consider the *cpdf* = $f(z_1, z_2, ..., z_n | z_0)$, where z_0 denotes the conditioning data at n_0 locations. This probability function can be defined as

$$f(z_1, ..., z_n | z_0) = f(z_1 | z_0) \cdot f(z_2 | z_1 \cup z_0) ... \\ \cdot f(z_n | [z_1 ..., z_{n-1}] \cup z_0) \quad (3)$$

Thus the generation of a realization by *Sequential Simulation* takes the following steps [Christakos, G. (2005)]:

(1) Draw a value z_1 from the conditional probability distribution f_1 given the set z_0 as conditioning data.

(2) Draw a value z_2 from the conditional probability distribution f_2 given $z_0 \cup z_1$ as conditioning data. … …

(n) Draw the last value z_n from the conditional probability distribution f_n given the set $z_0 \cup [z_1,...,z_{n-1}]$ as conditioning data.

Remark 2: the Sequential Simulation is conditional by construction, thus eliminating the extensive conditioning steps required by other traditional methods such as the *Turning Bands Method*.

Remark 3: There is no restriction on the spatial locations of the random variables yielding an algorithm that can be equally applied to generate one or more variables on either a regular or irregular grid.

However, it remains the problem of determining the cumulative portability distribution function (***cpdf***) of any single random variable given any set of conditioning data. This problem has been solved for the Gaussian distribution, where the data first are transformed to the standard Gaussian values. Simple or Ordinary kriging is used to obtain estimates of the necessary conditional distribution defined by the only the two Gaussian parameters; namely its mean and variance. The simulations are then drawn randomly from this distribution using *inverse transform* method. Finally, the results of the Gaussian simulation are transformed back into the original data space.

The Gaussian Function

Gaussian Function is unique in geostatistics for its analytical simplicity and for being the limit distribution of many analytical theorems globally known as '*Central Limit Theorem*'. If the continuous phenomenon $\{Z(x), x \in D\}$ is generated by the sum of a number of independent sources $\{y_k(x), x \in D, k = 1,...,K\}$ with similar spatial distributions then the phenomenon can be modeled by a Multi-Gaussian RF model. Multi-Gaussian models are extremely congenial, well understood, and they have large record of successful applications [John Dolloff and Peter Doucette (2014)]. A random function is said to be Gaussian or Multi-Gaussian if any linear combination of its variables follows the Gaussian distribution [Vanhatalo, J. and Vehtari, A. (2008)],

$$Z(x) = \sum_{k=1}^{K} \lambda_k Y_k(x) \approx gaussian - function \quad (4)$$

In geostatistics, conditional simulation is used to estimate, by *Monte Carlo Methods,* complicated nonlinear functions that depend explicitly on *multivariate stochastic distributions* [Ripley B. (2008)]. When the simulation domain is discrete, a sequential procedure can be considered [Journel, A.G (1989)]. This consists of prescribing an arbitrary ordering of all of the points of the domain, and simulating each point in turn according to a *Conditional Gaussian* distribution given the generated values of all the previous points. In the case where the simulation domain is continuous, a '*parallel*' procedure is necessary such as the *Turning Bands Method,* or *LU Decomposition (Cholesky) Method*.

LU Decomposition (or Cholesky Method)

The suggested method considers one covariance matrix C of all data and grid locations to be generated and partitioned as follows:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \qquad (5)$$

Where C_{11} is the variance-covariance matrix between data points. $C_{12} = C_{21}^t$ is the covariance matrix between data points and grid points and C_{22} is the variance covariance matrix of grid points. If matrices C_{11} and C_{22} are symmetric and positive-definite then matrix C is also symmetric and positive definite and can be decomposed by Cholesky algorithm into lower part and upper part as follows,

$$C = L \cdot U = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \cdot \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \qquad (6)$$

Let the vector $W = [W_1 \quad W_2]^t$ be a vector of independent *Gaussian Random* numbers $N(0,1)$, where the length of W_1 is equal to the number of data points and W_2 is equal to the number of grid nodes.

Also, let the vector $z_{us} = [z_1 \quad z_2]^t$ be an unconditional simulation of the random function $Z(x), \forall x \in D$ at data points and grid nodes with the covariance matrix C .

Now if we set $y = L \cdot W$, we will find that

$$E(y \cdot y') = E(LWW^tU) = L \cdot E(WW^t) \cdot U = L \cdot I \cdot U = LU = C \qquad (7)$$

$E(WW^t) = I$ is the identity matrix , because W is a vector of independent random numbers $N(0,1)$. From (7) we see that the vector $y = L \cdot W$ is an unconditional simulation of the random function $Z(x)$ that leads to the conclusion $z_{us} = [z_1 \quad z_2]^t = y$. Now we can write

$$z_{us} = y \rightarrow \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} L_{11} \cdot W_1 \\ L_{21} \cdot W_1 + L_{22} \cdot W_2 \end{bmatrix} \qquad (8)$$

As we seek a conditional simulation of grid nodes, we can set $z_1 = z_{data}$, where z_{data} is the vector of actual data values and set $z_2 = z_{nodes}$, where $z_{nodes} = z_g$ is the vector of conditional simulation (at nodes) which correspond to the random vector W_2 only. On the other hand the vector W_1 is no more a random vector and should be replaced by the solution of the upper part of the system (8)

$$W_1 = L_{11}^{-1} \cdot z_1 = L_{11}^{-1} \cdot z_{data} \qquad (9)$$

Now replacing W_1 in the system (8) yields the sought conditional simulation,

$$z_{nodes} = L_{21} \cdot L_{11}^{-1} \cdot z_{data} + L_{22} \cdot W_2 \qquad (10)$$

Remark 1: if a grid node happens to coincide (or co-located) with any data point, then the point should be considered as data and must be unique (no duplication of data is accepted at any location).

Remark 2: the data vector $z_1 = z_{data}$ in expression (9) have to be transposed (or normalized) so that the random function $Z_d(x)$ is *Gaussian Random Function* $Z_d(x) \in N(0,1)$.

Remark 3: all covariance matrices C_{11} , C_{12} and C_{22} should use the same covariance function so that *Variogram* parameters are normalized i.e. $c_0 + b = 1$. Note that

$$E(z_d z_d^t) = C_{11}, E(z_g z_g^t) = C_{22},$$
$$E(z_d z_g^t) = C_{12} = C_{21}^t \qquad (11)$$

Remark 4: multiple simulations may be generated easily and as many as desired (because the vector W_2 is a random vector), thus the matrix $L_{22} \cdot W_2$ can be computed in a single step. The number of this matrix rows will be equal to the number of simulated nodes and number of columns will be equal to the number of simulations. Again we do not need to generate W_1 because it already replaced by $L_{11}^{-1} \cdot z_d$

Cholesky Decomposition with Sparse Matrix Technique (the algorithm)

One can write Cholesky decomposition in the partition form (6) in different way. If we put

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \cdot \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \rightarrow$$
$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{bmatrix} \qquad (12)$$

Then:

- Compute all covariance matrices $\{C_{11}, C_{12}, C_{22}\}$ and store in sparse format.
- Compute the Cholesky decomposition of the square matrix C_{11} and obtain the sparse lower triangular matrix L_{11} as $C_{11} = L_{11}U_{11} = L_{11}L_{11}^t \qquad (13)$.
- Find the inverse of L_{11} and store it in L_{11}^{-1} . Note that we don't need to store U_{11} or U_{11}^{-1}
- Compute L_{21} using the formula:

$$L_{21} = C_{12}^{t} - \left(L_{11}^{-1}\right)^{t} \qquad (14)$$

Recalling from (12) :

$$C_{21} = L_{21} \cdot U_{11} \quad \Rightarrow$$

$$L_{21} = C_{12}^{t} \cdot U_{11}^{-1} = C_{12}^{t} \cdot \left(L_{11}^{t}\right)^{-1} = C_{12}^{t} \cdot \left(L_{11}^{-1}\right)^{t}$$

- Compute L_{22} : to find L_{22} we must find $(L_{22} \cdot U_{22})$ first. from (12) we have:

  $$L_{22} \cdot U_{22} = C_{22} - L_{21} \cdot U_{12} = LU_{22} \qquad (15)$$

  Then perform *Incomplete Cholesky Decomposition* of LU_{22} , which is also a sparse matrix and thus the output will be a sparse triangular matrix L_{22} .

Figures (2) through (9) show the sparse structure of the matrices used by the LU decomposition method. We can easily recognize the sparsity of each matrix visually. The sparsity of each matrix is computed by dividing the number of non-zero, which is fixed below each figure (nz~), by the total number of elements of the matrix.
Figure (4) through (9) below show the sparse structure of the matrices for larger problem, where the range of influence (the *Variogram* reaches the sill) having the value equal to the third of the maximum distance in the grid system. The figures show that the sparsity becomes much clear. The non-zero elements related to the total number of elements especially for the initial matrices are less than 5%. Although each element needs 8 MB for storage, there is much saving in the processing time as well as in storage capacity required for completing the simulation than that with known traditional method.

Implementation the Algorithm

- Dataset that has been used for testing programs performance (which cover geographic area of $1000m \times 1000$m) is given in the form of 3-column matrix (x, y, z) . It is a terrain elevation data consist of 266 points distributed as shown in (Fig.1).

- Data in the study has been downloaded from internet which was related to a small forested area in Wisconsin, USA, provided by Department of Forest Resources, University of Minnesota.

  The Conditional simulation by Incomplete Cholesky decomposition using sparse matrix technique has been implemented using the special Matlab Program.

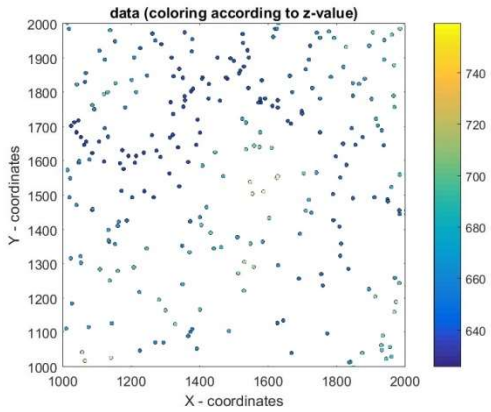

Fig.1 Dataset Locations and their distribution

The data is an ascii file, where Matlab reads it in two ways: either by giving the name of the 'file' or matrix which must consist of three columns: the first two columns contain the geographic x,y coordinates and the third contains the corresponding z values. The other way of entering data into the program is to give the names of 3 vectors, representing the geographic coordinates and data values, separately. The program computes the mean value and the variance in order to convert the data into a standard Gaussian (Davis 1987b). The second step is to define the grid system that has to be simulated. The parameters for the simulated nodes are entered in two way either interactively or written directly in the program. Variogram parameters, Anisotropy, nugget effect, number of simulations as well as seed number all can be entered in the same ways mentioned above. The program structure is similar to the program ***'lusim'*** provided by the GSLIB [Deutsch C.V. & Journel AG. (1992)], although here the study uses sparse matrix technique with the *Incomplete Cholesky Decomposition*. All those functions are Matlab built-in functions, thus they do computational tasks, much faster than other functions that have no similar Matlab functions. Those functions use the traditional GSLIB methodology and their execution is very slow, therefor they slow the performance of the program. For example, the construction of covariance matrices uses the traditional method and takes more than 90% of the overall execution time. 24 Simulations were generated and some results of the Cholesky decomposition Process are shown in figures No 2 through figure No 9 for small scale problems as well as for large scale Problems. Final 16 Simulation results represented by Contour images are shown in figures No.10 through figure. No.25.

Conclusions

In this paper, sparse matrices technique with Cholesky decomposition has been tested and proved as an efficient method for decomposing large covariance matrix by Cholesky method and generating simulation realizations. This method is based on the randomized sampling of covariance matrix for finding a sparse matrix which has much smaller size than the original one and captures most of the action of that matrix. This method works very well for approximation of DEM's which generates as many simulations as needed very fast.

When the field correlation is defined using *Gaussian Covariance Function* and taking into account the sparsity of the system and this means that only pair of nodes that fall within the zone of influence (the range) have a significant correlation, the rest of pairs, usually located beyond the search radius, having negligible correlations and introducing many zero elements in the covariance matrix. This study shows that, using this method, very large covariance matrices can be decomposed, but only limitation of this method is related to storing a large sparse matrices in computer memory like matrices L_{11} and L_{22}. The study provided computationally efficient methods for fitting DEM model to a relatively small data set by generating spatial simulations conditioning on the data itself. Once the new value simulated, it is added to the original set of conditioning data, and the procedure repeated. Once enough simulations are computed, a 'best' DEM model is then fit very quickly. The conditional simulation results give the most likely values or expected values at unobserved locations. As we see from the figures below that the simulated data reflect some of the uncertainties that are expected from any kind of simulation whether it is conditional or unconditional.

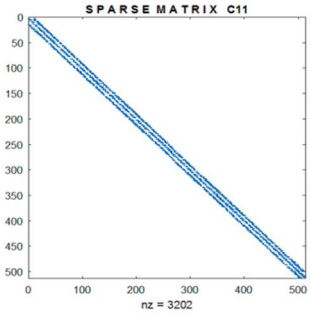

Figure 4 Sparse Covariance Matrix C11

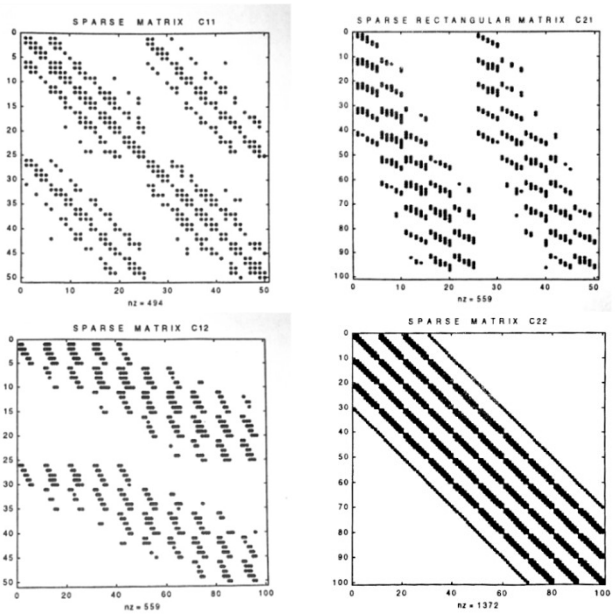

Figure 2 Computational Steps of the of the Variance Covariance Matrix (small scale dataset)

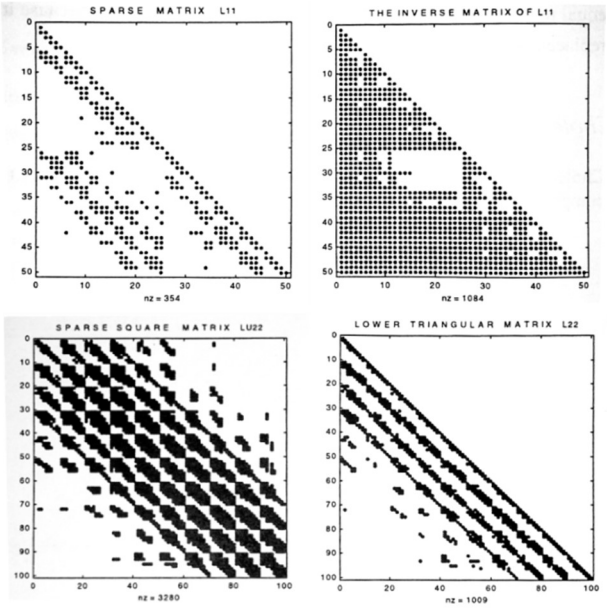

Figure 3 Computation Steps - Incomplete Cholesky Decomposition with Sparse Matrices (Small Scale Problem)

Figures (4) through (9) show***: Computational Steps***– Variance-Covariance Matrix and Incomplete Cholesky Decomposition with Sparse Matrices (for Large Scale Problem)

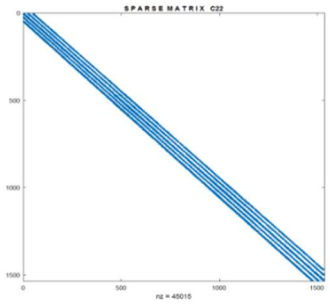

Figure 5 Sparse Covariance Matrix C22

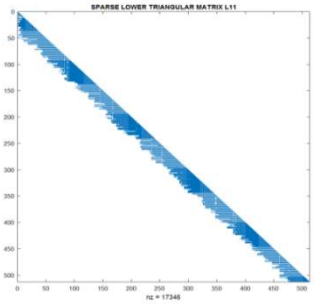

Figure 6 Sparse Lower Triangular Matrix L11

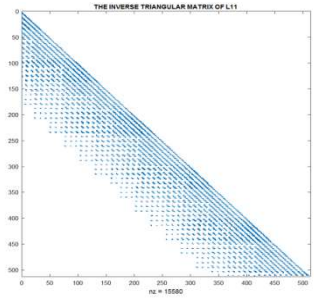

Figure 7 The Inverse Triangular Matrix of L11

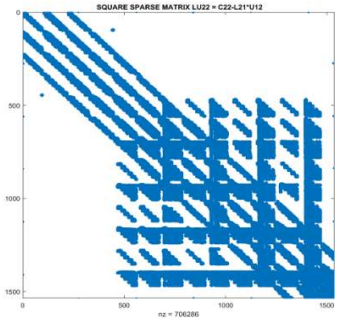

*Figure 8 Square Sparse Matrix LU22=C22-L21*U12*

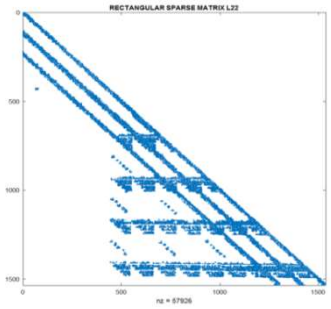

Figure 9 Rectangular Sparse Matrix L22

Some Simulations and Contours representation

Below12 figures represent 12 Simulation (Fig. No.10 through Fig.No.21) . Notice that each simulation is different from the others. In fact we can do unlimited number of simulations and each of them will be unique.

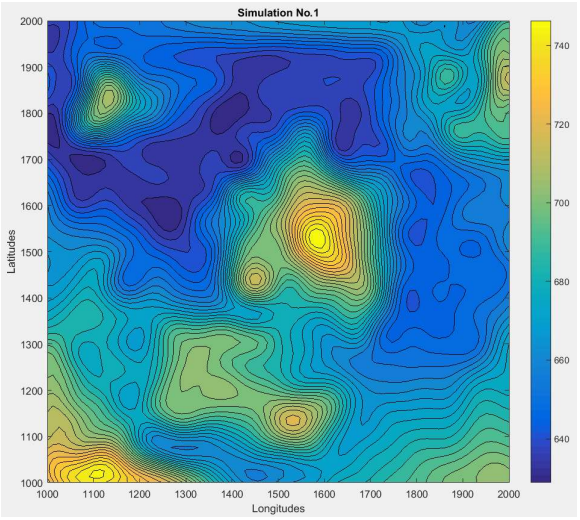

Fig.10 Simulation1

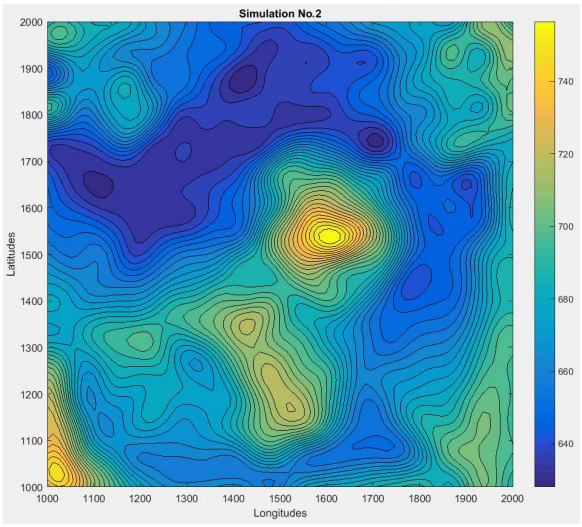

Fig.11 Simulation2

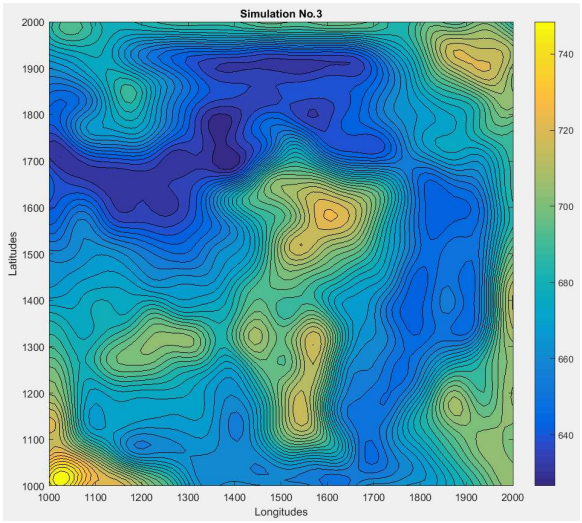

Fig.12 Simulation 3

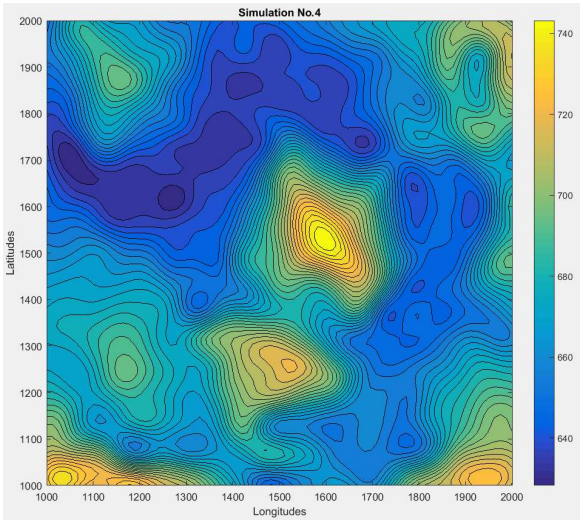

Fig.13 Simulation4

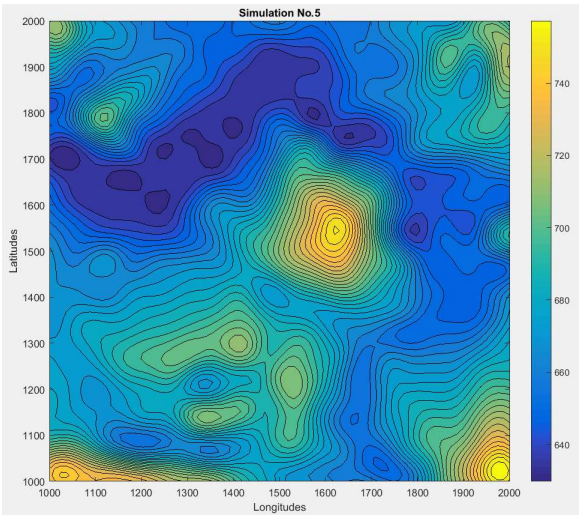

Fig.14 Simulation 5

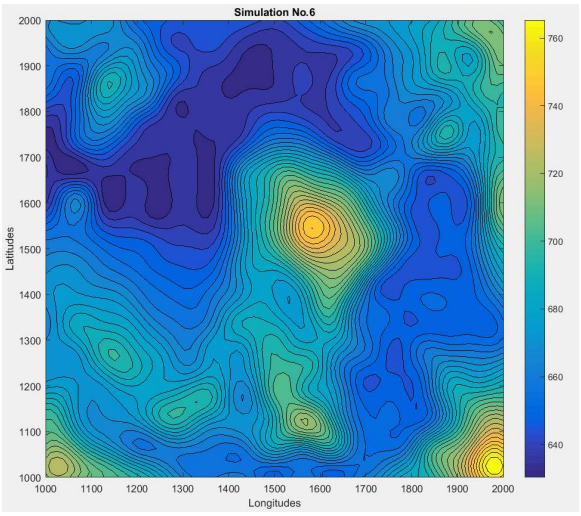

Fig.15Simulation 6

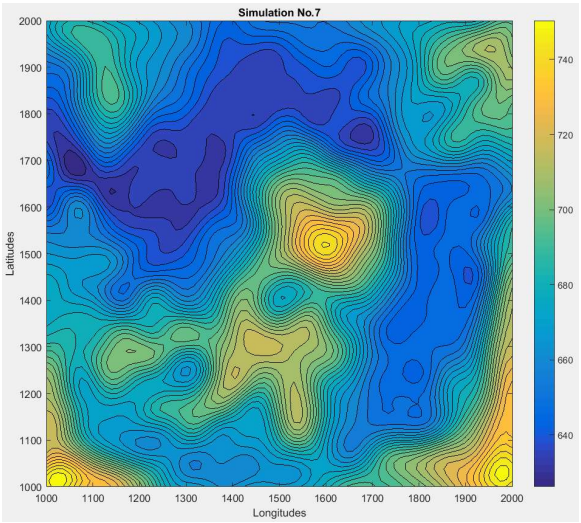

Fig.16 Simulation 7

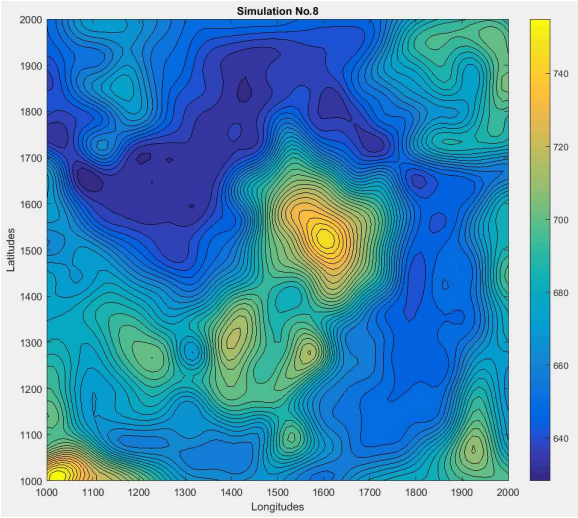

Fig.17 Simulation 8

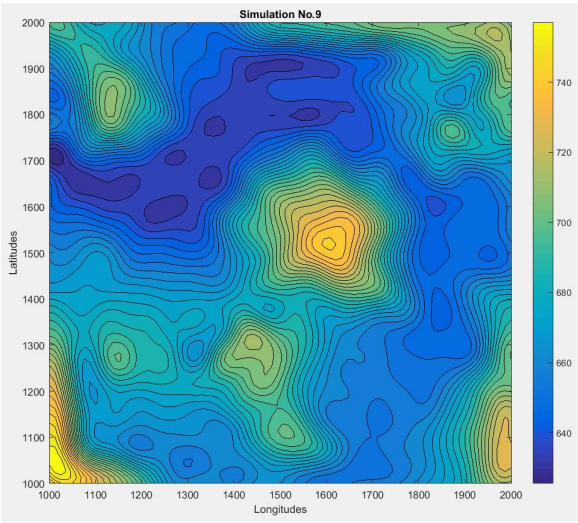

Fig.18 Simulation 9

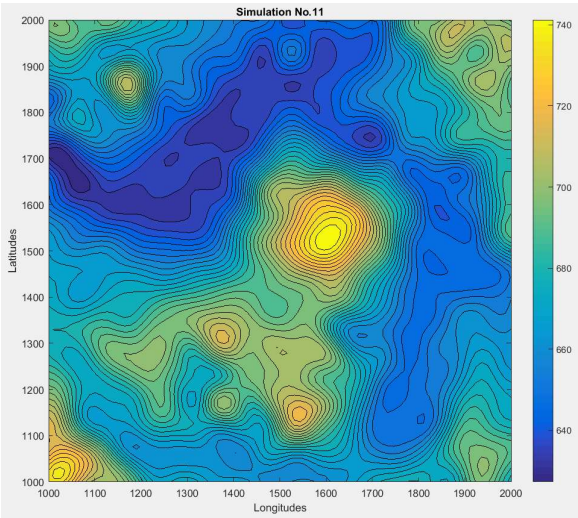

Fig.19 Simulation 11

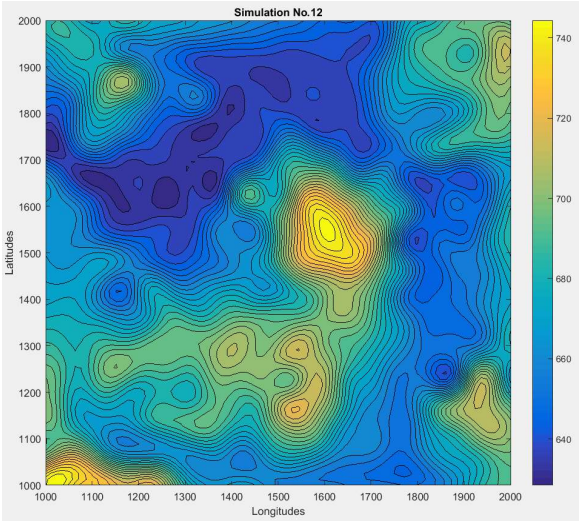

Fig.20 Simulation 12

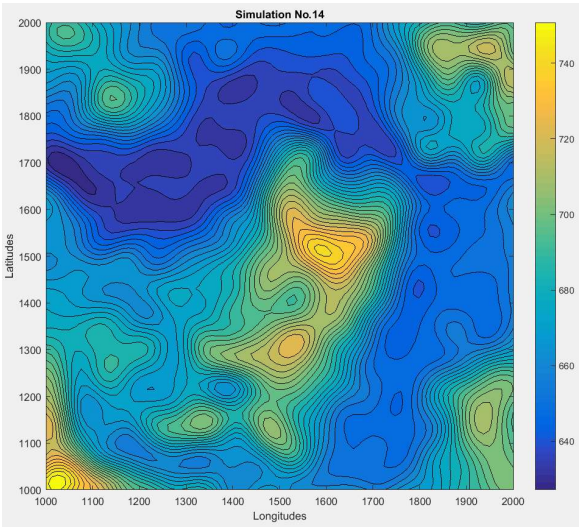

Fig.21 Simulation 14

REFERENCES

Alexander Malinowski, Martin Schlather, Peter J. Menck (2015) Analysis, Simulation and Prediction of Multivariate Random Fields with R-Random Fields.

Atkinson P, Quattrochi DA, Goodman HM (2000) Introduction to geostatistics and geospatial techniques in remote sensing.

Banerjee S. (2004) On Geodetic Distance Computations in Spatial Modeling.

Banerjee S, et al (2008) Gaussian predictive process models for large spatial data sets.

Bolstad W. M. (2007) Introduction to Bayesian Statistics. 2nd Edition.

Brus DJ, Heuvelink GBM (2007) Optimization of sample patterns for universal kriging of environmental variables.

Chiles, J.; Delfiner, P. (1999) Geostatistics: Modeling Spatial Uncertainty; Wiley.

Christakos, G. (2005) Simulation of Natural Processes. In Random Field Models in Earth Sciences; Dover: New York; pp. 295–336.

Cressie, N. (1985a) Fitting Variogram Models by Weighted LeastSquares(Mathematical Geology) p.563-586.

Cressie, N.A.C. (1993). Statistics for Spatial Data. Wiley.

Cressie, N.; Wikle, C.K. (2011) Statistics for Spatio-Temporal Data; Wiley.

David P. Landau, Kurt Binder (2009) A Guide to Monte Carlo Simulations in Statistical Physics (3rd Edition)

David O'Sullivan and David J. Unwin (2010) Geographic Information Analysis 2nd Ed.

Davis, J .C., (1986) Statistics and Data Analysis in Geology.

Davis, MW (1987a) Production of Conditional Simulation via the LU Triangular Decomposition of the Covariance Matrix (Mathematical Geology)

Davis, MW (1987b) Generating Large Stochastic SimulationsThe Matrix Polynomial Approx. Method

Deutsch C.V. & Journel AG. (1992) GSLIB, Geostatistical Software Library and User's Guide.

Dietrich, C. R. (1993). Computationally efficient Cholesky factorization of a covariance matrix with block Toeplitz structure.

Dietrich, C. R. and Newsam, G. N. (1997). Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix.

John P. Wilson (2012) Digital terrain modeling.

Journel, A.G. & Huijbregts, C. (1978) Mining Geostatistics.

Journel, A.G (1989) Fundamentals of Geostatistics.

Gomez-Hernandez, J .J . & Cassiraga E.F. (1994) Theory and Practice of Sequential Simulation (Workshop on Geostatistical Simulation, France,l993)

Goovaerts, P. (1997) Geostatistics for Natural Resources Evaluation; Oxford University Press.

Gneiting Tilmann et al. (2005), Fast and Exact Simulation of Large Gaussian Lattice Systems in Tech-Report no.477

Hartikainen, J. and Sarkka, S. (2010). Kalman filtering and smoothing solutions to temporal Gaussian process regression models

Hengl T. (2007) A Practical Guide to Geostatistical Mapping of Environmental Variables

Isaak, E.H. & Srivastava RM. (1989) Applied Geostatistics.

John Dolloff and Peter Doucette (2014) The Sequential Generation of Gaussian Random Fields for Applications in the Geospatial Sciences.

Lantuejoul C. (2002) Geostatistical simulation; models and algorithms. Springer, Berlin

Møller, J. (Ed.) (2003) An introduction to model-based geostatistics.

Myers, J .C., (1997) Geostatistical error Management, Quantifying Uncertainty for Environmental Sampling and Mapping.

Mund Jan-Peter (2013) Geospatial statistics and spatial data interpolation methods.

Oksanen J. (2006) Digital Elevation Model Errors in Terrain Analysis, Helsinki. (PhD)

Ripley B.D. (2008) Stochastic Simulation .

Schabenberger P.O., Gotway C.A. (2004) Statistical Methods for Spatial Data Analysis. Simulation of Random Fields.

Vanhatalo, J. and Vehtari, A. (2008). Modelling local and global phenomena with sparse Gaussian processes.