

Preliminary Research Toward Direct Geostatistical Simulation of Unstructured Grids

Y. L. Xie¹ and C. V. Deutsch
Centre for Computational Geostatistics
University of Alberta, Edmonton, Alberta

T. T. Tran
Chevron Petroleum Technology Company
San Ramon, California

ABSTRACT

Unstructured grids are receiving more attention in reservoir modeling. This is due to the promise of more realistic representation of geological heterogeneities and more flexible adaptation to the non-uniform flow near well bores. Conventional geostatistical algorithms are for a regular Cartesian grid in stratigraphic coordinates. Simulation is used to assign either point estimates at the grid nodes or block estimates of equal-sized blocks on a regular grid. Directly assigning petrophysical properties on unstructured grids presents unique challenges such as (1) searching for nearby relevant data and previously assigned grid blocks, and (2) calculating the covariances between the data and grid blocks and between the grid blocks and themselves. These covariances are needed for kriging to get the expected value and variance of conditional distributions in a sequential simulation scheme.

The covariance between blocks requires calculation of volume averaged covariance or variogram functions instead of simple point variogram. In practice, these average variogram or “gammabar” values are calculated by numerical integration, which is very time-consuming due to the large number of blocks involved in a simulation and the high level of discretization required for reliable integration.

We propose a very fast method for gammabar calculation on unstructured grids. The key aspects (1) using a table lookup of the variogram function instead of using the equation each time, (2) exploiting symmetry in the numerical integration to reduce the number of calculations, and (3) approximating the gammabar values through a trained neural network that captures the complex nonlinear relationship between the gammabar values and the unstructured grid. Future issues related to the direct simulation of unstructured grid are summarized.

¹ Now with Pacific Northwest Laboratories in Richland Washington (yulong.xie@pnl.gov).

INTRODUCTION

Irregular grids are being considered more often in reservoir flow simulation [1]. Irregular grids provide flexibility to account for geological constraints associated with faults, stratigraphic pinch outs, lithofacies heterogeneities and flow constraints associated with well locations and interaction between injection and production wells [1,5,6,13]. Figure 1 shows some unstructured grids such as the so-called Tartan grid, corner point grid [4], and locally refined grid (LRG) [11]. A review of unstructured grids used in flow simulation is presented in a companion report [14].

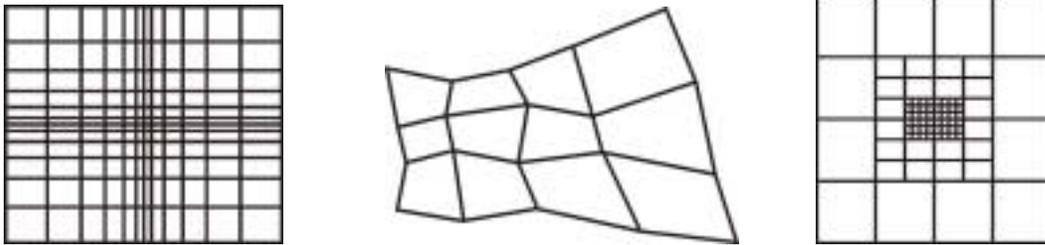


Figure 1: Some unstructured or irregular grids Tartan (*left*), Corner point (*middle*) and LRG (*right*)

Conventional geostatistical modeling is conducted on regular Cartesian grids with blocks of the same size and shape. The assumption is that all blocks have a constant volume support. In fact, the properties are typically assigned at grid node locations with the same volume as the data. There is no explicit handling of unstructured grids in geostatistical modeling. One evident approach is to construct a fine-scale geostatistical model and scale that model to the unstructured irregular grid system; however, this is inefficient particularly when multiple realizations are used for uncertainty assessment.

Direct population of irregular grid systems would be convenient, CPU efficient, and minimize storage requirements. The key is to use block Kriging as the basis of most geostatistical estimation and simulation. The Kriging mean and variance are important parameters for stochastic simulation. The Kriging equations are general in the sense that there are no constraints on the volume support of the input data. The block estimate on unstructured grids can be written as follows:

$$Z^*(v(\mathbf{u})) = \sum_{\alpha=1}^n \lambda_{\alpha} Z(v(\mathbf{u}_{\alpha}))$$

$$\sigma_k^2 = \bar{C}(v(\mathbf{u}), v(\mathbf{u})) - \sum_{\alpha=1}^n \lambda_{\alpha} \bar{C}(v(\mathbf{u}_{\alpha}), v(\mathbf{u}))$$

$$\begin{pmatrix}
 \bar{C}(v(u_1), v(u_1)) & \bar{C}(v(u_1), v(u_2)) & \dots & \bar{C}(v(u_2), v(u_{n-1})) & \bar{C}(v(u_2), v(u_n)) \\
 \bar{C}(v(u_2), v(u_1)) & \bar{C}(v(u_2), v(u_2)) & \dots & \bar{C}(v(u_2), v(u_{n-1})) & \bar{C}(v(u_2), v(u_n)) \\
 & & \dots & & \\
 \bar{C}(v(u_{n-1}), v(u_1)) & \bar{C}(v(u_{n-1}), v(u_2)) & \dots & \bar{C}(v(u_{n-1}), v(u_{n-1})) & \bar{C}(v(u_{n-1}), v(u_n)) \\
 \bar{C}(v(u_n), v(u_1)) & \bar{C}(v(u_n), v(u_2)) & \dots & \bar{C}(v(u_n), v(u_{n-1})) & \bar{C}(v(u_n), v(u_n))
 \end{pmatrix}
 \begin{pmatrix}
 \lambda_1 \\
 \lambda_2 \\
 \dots \\
 \lambda_{n-1} \\
 \lambda_n
 \end{pmatrix}
 =
 \begin{pmatrix}
 \bar{C}(v(u_1), v(u)) \\
 \bar{C}(v(u_2), v(u)) \\
 \dots \\
 \bar{C}(v(u_{n-1}), v(u)) \\
 \bar{C}(v(u_n), v(u))
 \end{pmatrix}$$

Block covariances (average variogram or *gammabar*) values are required instead of point variogram values in order to assign properties on unstructured grids. Second, a modified search strategy must be used to identify those blocks that are related to the block being considered. Third, when considering soft data, cokriging and the corresponding cross average variogram values have to be evaluated.

Average variogram or *gammabar* $\bar{\gamma}(v,v)$ and $\bar{\gamma}(v,v')$ values are calculated numerically by averaging the point variogram values of data pairs discretizing the block volumes. It is assumed that averaging is linear. The numerical evaluation of *gammabar* is a CPU-expensive process that is a bottleneck for the practical usage of unstructured grid kriging methods; therefore, we focus on the fast calculation of *gammabar* values.

BACKGROUND ON UNSTRUCTURED GRIDS

Regular Cartesian coordinate grids after stratigraphic coordinate transformation are the most widely grid systems for reservoir modeling and flow simulation. These grids are simple and the stratigraphic coordinate system transformation makes the grid conform to reservoir stratigraphic boundaries [2]; however, there are drawbacks [1].

Grid orientation effects [10] are an often-mentioned problem with regular Cartesian grids in flow simulation. In a simulation study of unfavorable mobility displacement, the areal orientation of the grid system, with respect to the location of the injectors and producers, can have a large affect the calculated results. Significant *numerical dispersion* [12] is caused by a combination of lower-order differencing schemes with large block size and a grid geometry that does not conform locally to the geometry of the flow field.

Numerous studies have been conducted toward alleviating the difficulties with conventional Cartesian grid [1,5,6,13]. They include local refinement on the base grids [11], conforming mapping of Cartesian grid from computational domain into curvilinear grid in physical domain [3], and flexible gridding systems [9,13]. In the companion report, various gridding methods are reviewed and classified based on the nature in the construction and their pros and cons are summarized [14].

In fact, even geostatistical simulation with Cartesian coordinate systems has trouble conforming to large scale curvilinear structural heterogeneities [7]. The traditional rectilinear Cartesian coordinate axes cut through meanders and folds, irrespective of the original sequence of sedimentation/genesis. This results in sample variograms that are typically too noisy and interpolation schemes that do not respect actual geological continuity.

Even if the influence of gridding on geostatistical simulation is not as significant as in the flow simulation, it will be computationally efficient to conduct geostatistical simulation in the same gridding system as the flow simulation.

Numerical Evaluation of Gammabar

The average variogram between two block volumes can be calculated based on an integration formula. The integration is conducted over the two volumes. In practice, the numerical summation shown below closely approximates the analytical solution:

$$\gamma(v_1, v_2) = \frac{1}{v_1 v_2} \int_{v_1} du \int_{v_2} \gamma(u - u') du' \cong \frac{1}{n_{v_1} n_{v_2}} \sum_{i=1}^{n_{v_1}} \sum_{j=1}^{n_{v_2}} \gamma(u_i - u_j')$$

where $n_{v_1} = nx_1 \times ny_1 \times nz_1$ and $n_{v_2} = nx_2 \times ny_2 \times nz_2$

CPU time saving is possible by: (1) keeping n_{v_1} and n_{v_2} as small as possible, (2) calculating $\gamma(u_i - u_j')$ as fast as possible, and (3) calculating $\gamma(u_i - u_j')$ for as few data pairs as possible. We also consider approximating gammabar $\gamma(v_1, v_2)$ with the point variogram between the centroids, $\gamma(u_i - u_j')$, and a correction for the influence of the shapes, sizes, spatial orientations and separation of the two volumes.

Procedures aimed at speeding up the gammabar calculation are proposed. First, to avoid tedious direct calculation of variogram for every data pair discretizing the blocks, a variogram value lookup table is pre-calculated. In the evaluation of gammabar between two blocks, the variogram values of data pairs are retrieved from such a lookup table.

Second, the symmetry of the data pairs is used to reduce the number of pairs to be considered. The blocks are enclosed by circumscribed rectangles or rectangular parallelepipeds in 2D and 3D cases, respectively, and discretized with equal increments in each direction for both blocks. Only data pairs having unique variogram distances will be calculated and stored. If the blocks are rectangular the number of data pairs with the same unique variogram distance can be determined beforehand, which makes the gammabar calculation a summation of the unique variogram values multiplied by their number of data pairs. For blocks with irregular shapes, the accumulation of gammabar from the unique variogram values will be modified by the number of discretized data pairs inside the blocks.

Third, further speeding up in the gammabar evaluation may still be required with the consideration of many blocks in a grid system. The diagonal elements in the kriging matrix are critical and they are calculated precisely with the steps above. The off-diagonal elements are for volumes that do not overlap and could be approximated by correcting from the point variogram between the centroid of the blocks. Training data is generated for each major isotropic variogram model considering the possible distribution of the size, shape, orientation and separation of two blocks in the normalized range space. The ratio between the gammabar and point variogram is pre-calculated. The relationship between this factor and the size, shape, orientation, spatial separation of the two blocks is established by a neural network. For a block pair in a real situation, the point variogram between centroids will be calculated, a correction factor will be predicted, and the gammabar will be approximated by correcting the point variogram with the factor.

Numerical Integration for Gammabar

The discretizing level is one of the factors influencing the computation of gammabar. The precision of the approximation of the numerical solution to the analytical average variogram depends on the discretization level. Two arbitrary variogram models are used to give an idea about the asymptotic convergence of the gammabar value for increasing levels of numerical discretization.

$$\Gamma_1 = 0.6Exp \begin{pmatrix} -33.5/1000 \\ 21.6/3000 \\ 355/112 \end{pmatrix} + 0.4Sph \begin{pmatrix} 0.0/30000 \\ 0.0/6000 \\ 0.0/3350 \end{pmatrix}$$

$$\Gamma_2 = 0.4Spe \begin{pmatrix} -33.5/1000 \\ 21.6/3000 \\ 355/112 \end{pmatrix} + 0.3Exp \begin{pmatrix} -33.5/10000 \\ 21.6/4500 \\ 355/1350 \end{pmatrix} + 0.3Gau \begin{pmatrix} -33.5/30000 \\ 21.6/6000 \\ 355/3350 \end{pmatrix}$$

The two blocks are assumed to be rectangular parallelepipeds. The table below shows the gammabar values for different discretization levels.

Discretization Level	Number of data pairs	Average Variogram (Model 1)	Average Variogram (Model 2)
2	64	0.2505	0.1356
3	729	0.2700	0.1485
4	4,096	0.2716	0.1523
5	15,625	0.2773	0.1539
6	46,656	0.2784	0.1548
7	117,649	0.2791	0.1553
8	262,144	0.2795	0.1556
9	531,441	0.2798	0.1559
10	1,000,000	0.2800	0.1560
11	1,771,561	0.2801	0.1562
15	11,390,625	0.2804	0.1564
21	85,766,121	0.2806	0.1565
31	887,503,681	0.2807	0.1566

The calculated average variogram levels out when the discretization number exceeds 5 and approaches 10. Therefore, 10 may be the maximum number one wants to use for the discretization of a block in any one direction.

LOOKUP TABLE

The original gammabar program in GSLIB proceeds by (1) calculating squared anisotropic distance for each data pairs between the two blocks, (2) getting the distance by a square root, and (3) calculating the variogram based on the variogram formulas. The following formulas are used:

$$\begin{pmatrix} h_\alpha \\ h_\beta \\ h_\theta \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & \lambda_1 & & \\ & & \lambda_2 & \\ & & & \lambda_3 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} h_x \\ h_y \\ h_z \end{pmatrix} \text{ and } h^2 = h_\alpha^2 + h_\beta^2 + h_\theta^2, \quad h = \sqrt{h^2}$$

where h_x , h_y and h_z are the distances in natural coordinate system and h_α , h_β , h_θ are the distance after considering axis rotation ($r_{i,j}$'s) and anisotropy (λ 's).

$$\left\{ \begin{array}{l} \gamma_h^{Sph} = \begin{cases} C \left[1.5 \frac{h}{a} - 0.5 \left(\frac{h}{a} \right)^2 \right] & \text{if } h \leq a \\ C & \text{if } h > a \end{cases} \\ \gamma_h^{Exp} = C \left[1 - \exp \left(-\frac{h}{a} \right) \right] \\ \gamma_h^{Gau} = C \left[1 - \exp \left(-\frac{h^2}{a^2} \right) \right] \end{array} \right.$$

The direct calculation of variograms takes time, especially for the exponential and Gaussian variogram models because of the exponentiation.

The idea is to pre-calculate variogram values up to certain distance limit for each variogram structure and to establish a variogram value lookup table. The direct calculation of the variogram value will be replaced by (1) calculating squared variogram distance for the data pairs between the two blocks, (2) getting the index of such a squared distance in the lookup table, and (3) retrieving the variogram value from the table. The lookup table is organized with equal squared distance to save the square root operation.

The time for direct calculation of the variogram is variogram model dependent. Exponential and Gaussian variogram models take much more time than Spherical variogram model. The table lookup depends on the squared distance calculation and indexing, which is independent of the variogram model type.

The time for table lookup and direct calculation is compared. 10001 points are kept in the lookup table, but the size of the lookup table has little influence on the time for retrieval. The following table is for a modest PC for calculation of 10 million variogram values. The time saving depends on the variogram types and nested structures in the variogram model.

	Table Lookup(s)	Call Cova3 (s)
Sph	4.2	14.3
Exp	4.2	21.2
Gau	4.2	22.4
Sph+Exp	7.4	54.4
Sph+Gau	7.4	56.7
Exp+Gau	7.4	70.1
Sph+Exp+Gau	10.6	123.0

The upper limit of the squared variogram distance in the lookup table has to be larger than the largest possible squared variogram distance. The upper limit can be set large enough to ensure complete coverage. A reasonable upper limit is to use the physical maximum separation between the extreme points of any two blocks.

EXPLOIT SYMMETRY

The original gammabar program enumerates all data pairs in the two blocks. The calculation is very heavy due to the number of the large number of data pairs; however, when the two blocks are discretized with the same increments, there are many duplicate calculations because of pairs with the same separation angle and distance. Figure 2 shows the duplication due to the symmetry.

It is noticed that the variogram between data pairs 2-5, 3-6, 4-6, 4-7, 5-8, 6-9 will be the same as that from data pair 1-4 due to the symmetry.

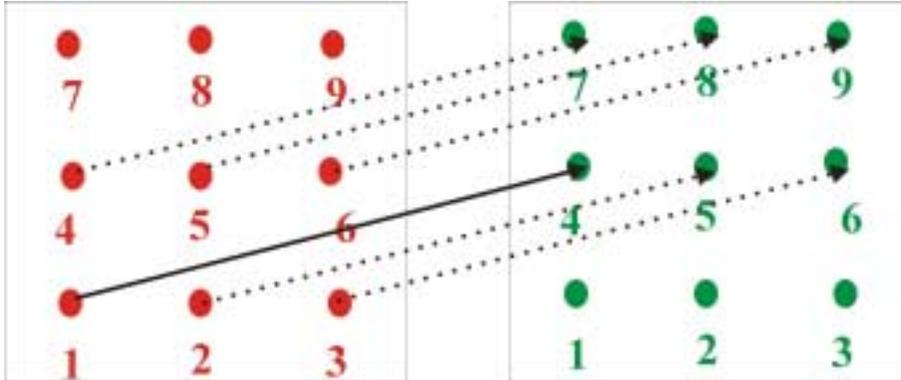


Figure 2. Illustration of the duplication of variogram distance.

Therefore the variogram value will be calculated *only* for data pairs with unique variogram distances. The number of data pairs for a certain unique variogram distance can be easily determined, which speeds the gammabar calculation.

Let nx_1, ny_1, nz_1 and nx_2, ny_2, nz_2 be the discretization number for the two blocks based on the same increments, dx, dy and dz . The data pairs with unique variogram distance only consist of (1) every point of the first Y-Z slice in block one (*red cells*) with every point in block two (*blue cells*), and 2) the first Y-Z slice in block two (*blue cells*) with every point in block one except those in the first Y-Z slice (*red cells*), as shown in Figure 4.

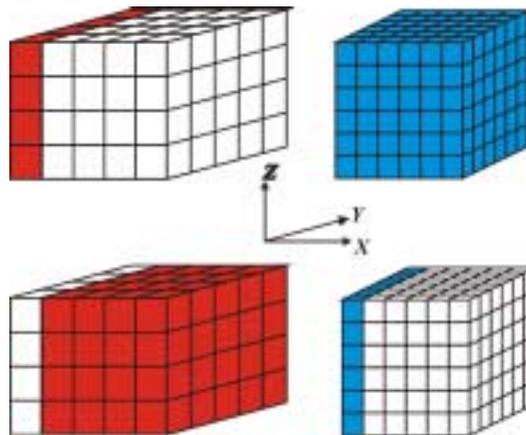


Figure 3. Data pair with unique variogram distance

The variogram values will be calculated only for those data pairs with unique variogram distances. These variogram values could be organized into a 3D tensor consisting of two

portions of dimension $nx_2 \times (ny_2 \times nz_2) \times (ny_1 \times nz_1)$ (denoted as A) and the first $Y-Z$ block in point of volume v_2 with every point in volume v_1 except those in the first $Y-Z$ block results a cube with a dimension of $(nx_1 - 1) \times (ny_2 \times nz_2) \times (ny_1 \times nz_1)$ (denoted as B).

For the regular blocks shown in Figure 3, the number of data pairs having the same unique variogram distance (value) can be determined and the gammabar value will be calculated by weighting the unique variogram values by the right number of the data pairs. The following pseudo code shows the use of the multiplication factor for each unique data pair to complete the gammabar calculation.

In the case of $nx_1 \leq nx_2$:

```

do ix=1,nx2
  if (ix ≤ (nx2-nx1+1))
    every element in A will be multiplied by nx1
  else
    every element in A will be multiplied by (nx2 - ix + 1)
  end if
end do
do ix=2,nx1
  every element in B will be multiplied by (nx1 - ix + 1)
end do

```

And in the case of $nx_2 > nx_1$:

```

do ix=1,nx2
  every element in A will be multiplied by (nx2 - ix + 1)
end do
do ix=2,nx1
  if (ix ≤ (nx1-nx2+1))
    every element in B will be multiplied by nx2
  else
    every element in B will be multiplied by (nx1 - ix + 1)
  end if
end do

```

Note that A and B are not saved since the multiplication can be done while the elements are being calculated. The number of data pairs to be calculated is: $nx_2 \times (ny_1 \times nz_1) \times (ny_2 \times nz_2) + (nx_1 - 1) \times (ny_1 \times nz_1) \times (ny_2 \times nz_2)$. The reduction factor in the number of data pairs to be conducted variogram calculation is:

$$\begin{aligned}
& \frac{nx_2 \times (ny_1 \times nz_1) \times (ny_2 \times nz_2) + (nx_1 - 1) \times (ny_1 \times nz_1) \times (ny_2 \times nz_2)}{(nx_1 \times ny_1 \times nz_1) \times (nx_2 \times ny_2 \times nz_2)} \\
&= \frac{(nx_1 + nx_2 - 1) \times (ny_1 \times ny_2 \times nz_1 \times nz_2)}{(nx_1 \times nx_2) \times (ny_1 \times ny_2 \times nz_1 \times nz_2)} \\
&= \frac{nx_1 + nx_2 - 1}{nx_1 \times nx_2}
\end{aligned}$$

The distinction between x , y , and z is arbitrary; therefore, it is always possible to choose the directions to ensure the greatest reduction in the data pairs.

The Table below shows a comparison between the modified and the original variogram program. The same two variograms models as listed above were used in the calculation. The CPU time and the average variogram values are listed.

Discretization No	Variogram Model 1				Variogram Model 2			
	Original Program		Modified program		Original Program		Modified program	
	Value	Time (s)	Value	Time(s)		Time (s)	Value	Time(s)
2	0.2505				0.1356			
3	0.2700				0.1485			
4	0.2716				0.1523			
5	0.2773	0.06	0.2770	0.06	0.1539	0.16	0.1538	0.06
6	0.2784				0.1548			
7	0.2791				0.1553			
8	0.2795				0.1556			
9	0.2798				0.1559			
10	0.2800				0.1560			
11	0.2801	7.47	0.2797	1.10	0.1562	11.15	0.1559	1.38
15	0.2804	44.71	0.2799	3.29	0.1564	68.72	0.1562	4.56
21	0.2806	333.73	0.2801	15.54	0.1565	512.84	0.1563	22.63
31	0.2807	3435.26	0.2802	107.44	0.1566	5298.49	0.1564	157.42

Computer configuration: P III 450, 128 MB RAM

The CPU time is also compared for the following four situations (1) enumerate all data pairs and direct calculation of variogram value by calling cova3, (2) enumerate all data pairs and retrieve variogram values from lookup table, (3) calculate only the unique data pairs and direct variogram calculation by calling cova3, and (4) calculate only the unique data pairs and retrieve variogram values from lookup table. The variogram model is the second variogram model listed in above and the discretization number is 21 for each lateral direction. The Table below gives the results. The lookup table saves about 50% CPU time and the reduction of data pairs saves 90% CPU time. The combination reduces the CPU to 5% of the original CPU time. The time saving is variogram model (shape and nest structure) dependent.

	Enumerate+Direct	Enumerate+Retrieve	Unique+Direct	Unique+Retrieve
Time (s)	492	231	47	22
Value	0.1565	0.1563	0.1565	0.1563

Note that the difference between 492 s (here) and 512 s (above) is due to two different programs. The latter is the original gammabar program and the former comes from the modified program.

Geometry Utilization for Irregular Blocks

A “large” rectangle is used to enclose irregular blocks. Two different irregular shapes, polygon (polyhedron) and ellipse (ellipsoid), are considered here.

Figure 4 shows two ellipses of different size and spatial orientation. The discretization increment in x direction is based on ellipse 1 which has a longer x side and the y increment is based on ellipse 2 which has a longer y side. The y origin of the circumscribed rectangle for ellipse 1 and the x origin of the circumscribed rectangle for ellipse 2 are shifted a little bit to let the centroids of the circumscribed shapes coincide with the ellipses.



Figure 4: The discretization of irregular blocks

As with regular shaped volumes, the variogram values for unique data pairs are calculated. The number of data pairs for each unique variogram distance is calculated by using an indicator for each discretized point in the block.

The indexing described below will allow quick retrieval of the corresponding unique variogram for each data pair. Once again, consider the discretization for two blocks as: nx_1, ny_1, nz_1 and nx_2, ny_2, nz_2 and the looping is on x direction. The entire unique variogram tensor is shown in Figure 5.

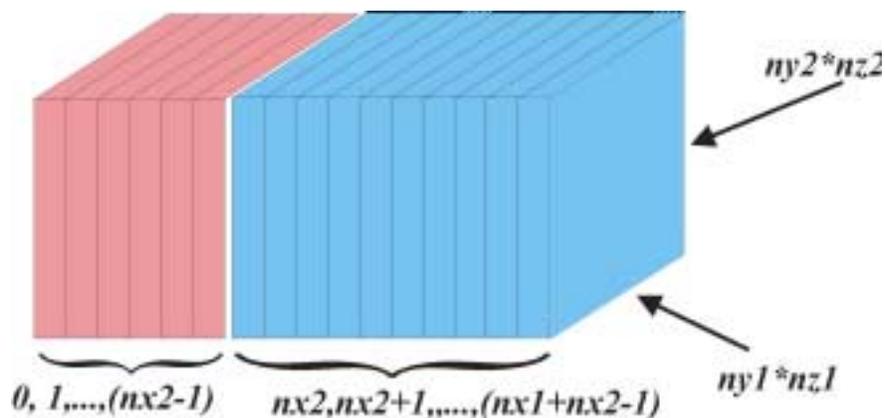


Figure 5: Tensor of unique variogram values

If we index the unique tensor in the way shown in Figure 5, we can then calculate and apply the unique variogram in the way shown as in the pseudo code as follows:

```

for ix=1:nx1
for jx=1:nx2
if (jx>=ix),
index = jx-ix
else if (jx<ix)
index = ix-jx+(nx2-1)
end if
end
end
end

```

Specifically:

1. if $(ix==1)$ or if $((ix>1) \& (jx==1))$ the gammabar should be calculated and stored in the indexed slice of the tensor
2. for all other ix and jx , the gammabar would be retrieved from the indexed slice in the tensor, *i.e.*:

$$\gamma = \frac{1}{n_{in}} \sum_{i=1}^{n_{in}} ind1 \times ind2 \times \gamma_{index}$$

$$\text{where } n_{in} = \sum_{i \in \text{block1}} \sum_{j \in \text{block2}} ind_i \times ind_j$$

The time improvement in the gammabar calculation for irregular blocks depends on the shape, size, irregularity, and spatial orientation.

Two block pairs have been used for time comparison. One pair consists of two rectangles with the coordinates of $\{25/25, 25/50, 50/50, 50/25\}$ and $\{25/50, 25/75, 50/75, 50/50\}$. The other pair consists of two polygons both having four vertices with the coordinates of $\{26/70, 30/90, 50/95, 54/65\}$ and $\{30/90, 26/105, 46/110, 50/95\}$. Variogram model two is used. The gammabar calculation is evaluated with the proposed procedure and compared with a reference method without considering the lookup table and the symmetry. The calculated gammabar values and the time are listed below. We see a significant saving in the computation.

	γ	$\bar{\gamma}$	$\bar{\gamma}'$	No	$T(s)$	$T'(s)$
Rectangles	0.0554	0.0557	0.0558	256	0.0050	0.050
	0.0554	0.0557	0.0558	1464	0.110	0.280
	0.0554	0.0557	0.0558	19448	0.280	2.690
	0.0554	0.0569	0.0570	923521	0.710	11.10
	0.0554	0.0557	0.0558	2825761	1.60	32.25
Polygons	0.0481	0.0474	0.0475	84	0.060	0.056
	0.0481	0.0482	0.0483	4094	0.060	0.270
	0.0481	0.0484	0.0485	52920	0.220	1.920
	0.0481	0.0483	0.0484	251120	0.490	7.140
	0.0481	0.0483	0.0485	767514	1.10	21.03

γ : point variogram between centroids of the blocks

$\bar{\gamma}$: gammabar calculated from modified algorithm

$\bar{\gamma}'$: gammabar calculated from brute force reference method

No: overall valid pairs

$T(s)$: overall time for modified method

$T'(s)$: overall time for reference method

APPROXIMATE TECHNIQUE

Approximation of gammabar for non-overlapping blocks

Although we show significant improvement in the computation time for gammabar values, an unstructured grid could consist of millions of blocks. Further reduction in the computation time is investigated for practical application.

The gammabar value is the average of the variogram values of data pairs between two blocks. All data pairs differ by the difference between the centroids of the blocks. The relationship between gammabar and the variogram value between the centroids is complex and nonlinear, which depends on the sizes, orientation, shapes, and separation of the blocks. Qualitatively, this relationship is interpretable based on the variogram model but is too complex to be formulated quantitatively for prediction purpose. We train a neural network to learn such a relationship and the gammabar value is then predicted from the point variogram value (between block centroids) by correcting the influence of the shapes, size, spatial orientation of the blocks. Note that this technique is only for non-overlapping volumes.

As a first approximation we assume that the blocks can be represented by an ellipse/ellipsoid where the centroid of the ellipse/ellipsoid is that of the irregular shape and the ratio of the long and short axis of the ellipse/ellipsoid reflect the geometric orientation of the irregular shape. The orientations of the long/short axes of the ellipse/ellipsoid are determined by a principal component analysis of the discretized irregular shape. The ellipse/ellipsoid will have the same area/volume of the irregular shapes.

A training set is constructed by randomly generating a set of paired ellipses. The long axes, short axes, azimuth angles of both ellipses, and the separation in both x and y directions are used as training attributes for the neural networks. The point variogram between the centroids and gammabar values are calculated. The ratio between the gammabar value and the point variogram value is obtained and used as the target attribute for neural network training. For generality, all parameters and calculations are performed in normalized space, that is, with a standard isotropic variogram model with unit range. In the normalized space, the distributions for drawing the attributes are:

<i>Training Attribute</i>	<i>Distribution</i>
Azimuth angle of ellipse 1	uniform distribution between 0 to 360
Long axis of ellipse 1	exponential distribution with average of 1
Short/Long of ellipse 1	uniform distribution between 0 to 1
Azimuth angle of ellipse 2	uniform distribution between 0 to 360
Long axis of ellipse 2	exponential distribution with average of 1
Short/Long of ellipse 2	uniform distribution between 0 to 1
Separation of centroid in x direction	exponential distribution with average of 1
Separation of centroid in y direction	exponential distribution with average of 1

<i>Target Attribute</i>	
Ratio of gammabar and centroid variogram	

As we discussed previously, the diagonal elements of the covariance matrix are critical; therefore, for the blocks themselves we calculate the gammabar values precisely. The approximation with point variogram between centroid and neural network will be used for the off diagonal elements (non-overlapping) average variogram calculation. 100,000 non-overlapping ellipsoids have been generated for neural network training.

Figure 6 shows the histograms of the training parameters and Figure 7 show the histograms of the target attributes, which are the ratio between gammabar and centroid variogram for three types of variogram models (these figures have been put after the references). Virtually all of the factors are between 0.95 and 1.05, which makes prediction difficult in some senses, but easy in the sense that slight errors are not compounded in later calculations. The attempts to use all 100,000 data for training are not successful due to the distributions of the attributes. A reduced training set containing 8504 data is retrieved from the entire data set.

Multivariate Regression and Neural Network

A supervised learning algorithm is used to discern a relationship between two sets of attributes. There are many multivariate statistical techniques, such as multiple linear regression (MLR), regression techniques based on latent variables, principal component regression (PCR) and partial least square regression (PLS). When the intrinsic relationship is believed to be non-linear, non-linear modification to those techniques is required or technique dedicated to nonlinear relationship such as multivariate Adaptive Regression Splines (MARS) and neural networks (NN) could be used. Neural network with Back Propagation (BP) of error is a popular method [8]. Besides BP, there are Radial Basis Functions (RBF), Generalized Regression NN (GRNN) for supervised learning. We tried multivariate regression with non-linear terms and BP-NN to discern a relationship between volume characteristics and the correction factor. The principle of BP network and implementation details can be found in the vast literature [8] of neural networks.

For multivariate regression, we supplemented the linear terms with exponential terms and squared terms to account for the nonlinearity. Figure 8 shows the scatter plots between the predicted values versus the real target values in the training set for regression models with different nonlinear terms. Note the very low correlation coefficient that suggests simplistic multivariate regression does not work very well.

For the neural network training, the target values are normalized to 0.2 to 0.8 using the maximum and minimum values. All the training attributes are scaled in to 0/1 range. Parameters such as momentum, learning rate, and number of nodes in the hidden layer have been varied to get optimal training configuration. Figure 9 shows the scatter plots between the predicted values versus the real target values in the training set. The much higher correlation coefficient suggests neural network is superior for this kind of nonlinear relationship learning.

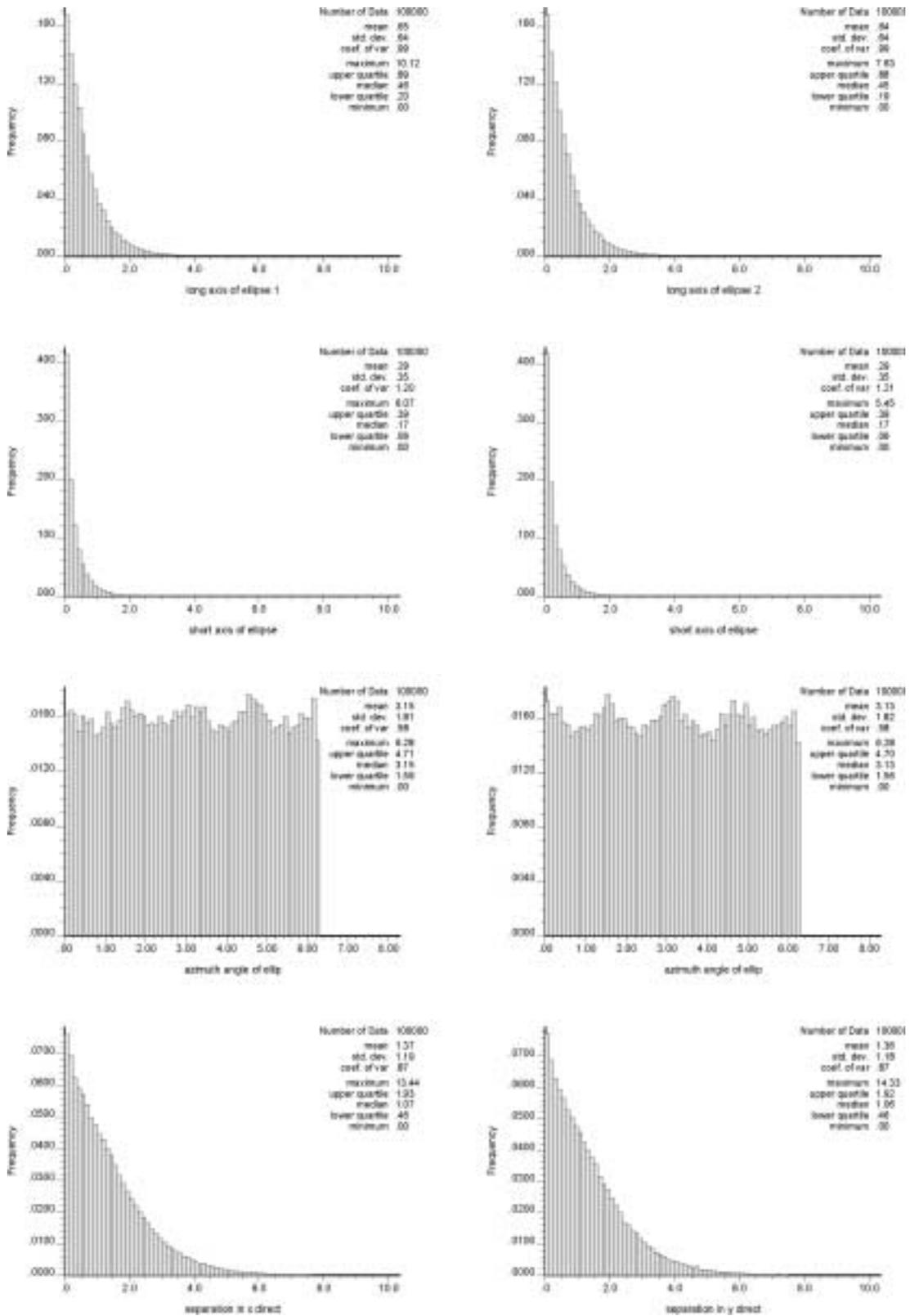


Figure 6 Histogram of the training parameters

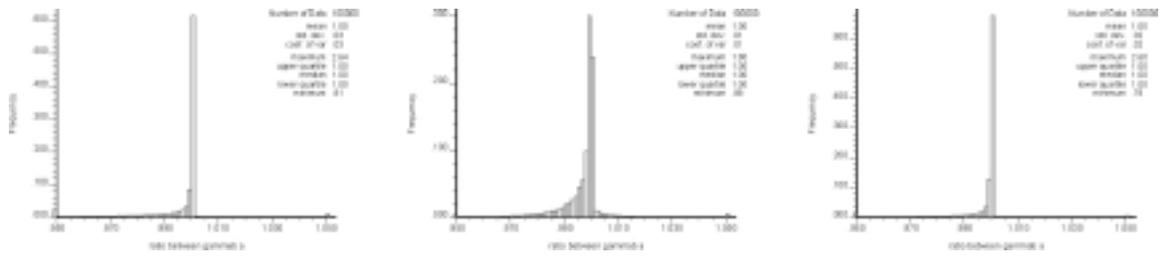


Figure 7. Histograms of the target attributes

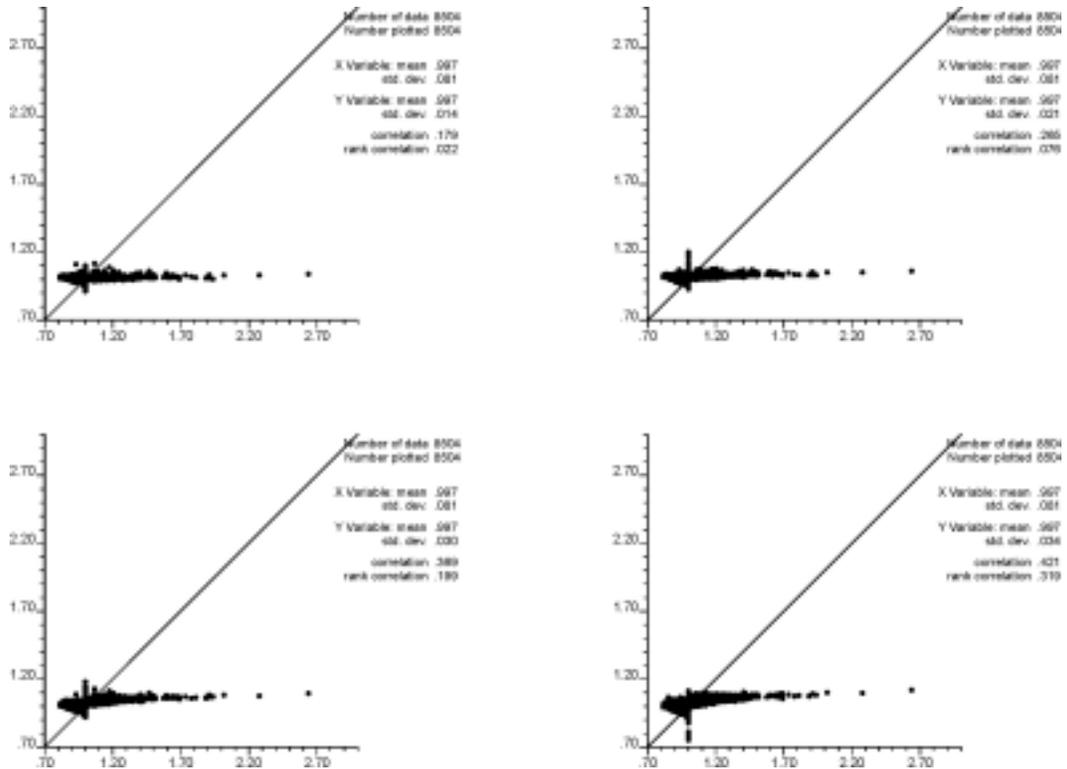


Figure 8 MLR training results

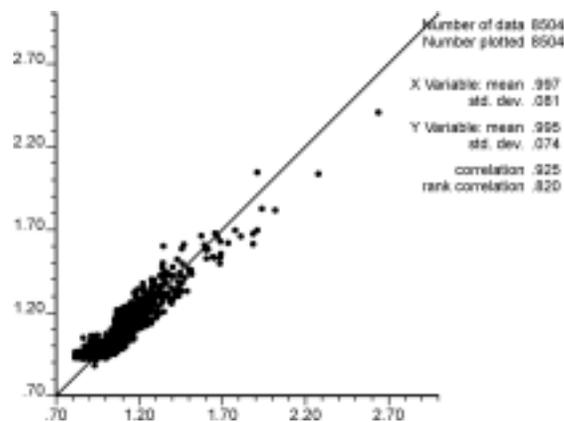


Figure 9 Neural network training result

Simulation Using the Approximated Covariance

A 16 by 16 2-D Tartan grid is shown in Figure 10. The size of the blocks reduces 500 by 300 to 125 by 75 in the center of the domain. A Spherical variogram model of

$$\Gamma_h = 0.10 + 0.90 \text{Sph} \left(\sqrt{\left(\frac{h_1}{1500}\right)^2 + \left(\frac{h_2}{500}\right)^2} \right) \text{ is assumed. Unconditional simulation is}$$

carried out in Gaussian space; we would expect to use Direct space in the future.

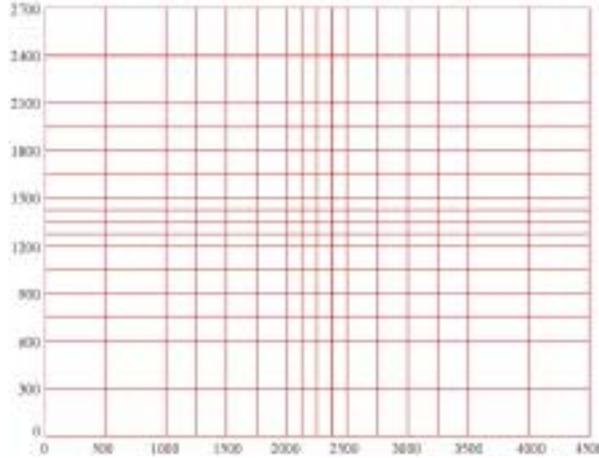


Figure 10: a 16 by 16 Tartan grid used for simulation

Four situations are considered for comparison: (1) using point covariance for covariance matrices in Kriging, (2) using precisely calculated block covariances, (3) using precisely calculated block covariance for diagonal elements and point covariance for the off-diagonal elements in the left hand covariance matrix, and (4) using precisely calculated block covariance for diagonal elements and approximated block covariance by trained neural network for the off-diagonal elements in the left hand covariance matrix of Kriging equations. The purpose of this investigation is to see if there are any problems such as singular matrices when the neural net approximation is used.

We found no singular matrices for any situation; however, we did notice some negative Kriging variances and unusually large Kriging weights (absolute value larger than 2) occurring when using point covariance or neural network predicted block covariance for the off-diagonal elements of the covariance matrix.

	<i>Reference Block Covariance</i>	<i>Point Covariance</i>	<i>Block/Point Covariance</i>	<i>Block/Approximated block covariance</i>
# Singular	0	0	0	0
. big weights	0	1	36	84
# neg. variance	0	0	18	18

These erratic large weights and negative Kriging variances have been investigated thoroughly. The negative variances are related to the unusual large weights. We checked each all 84 big weights and visualized the geometric configurations of the blocks

involved and found that the problem cases were almost always linked to some kind of screening. Figure 11 shows four such situations.

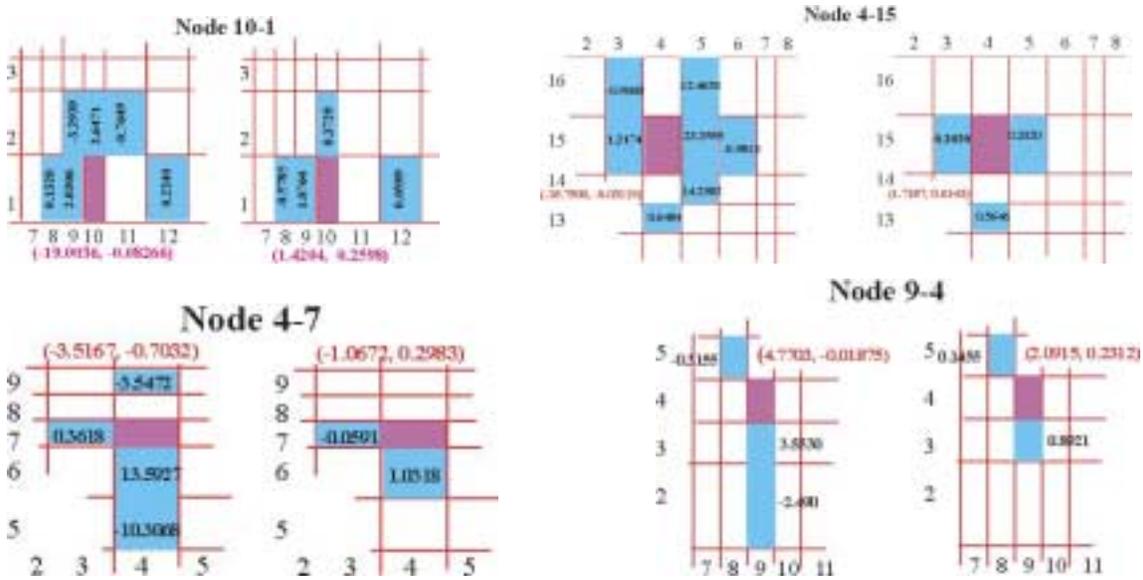


Figure 11 Four cases of the node configuration and weights of Kriging equations

Mostly, when two blocks are adjacent to the location being estimated, one block receives a large positive weight and the other block received a big negative weight for compensation. These cause the Kriged mean values and Kriging variance to be unreliable. Such problems associated with data screening effect become more serious as the simulation proceeds because there are an increasing number of nearby blocks and previous Kriging problems are compounded.

An iterative process to remove non-adjacent blocks from Kriging equation is proposed to solve this problem. For each Kriging system, if big weights occur to non-adjacent nodes ($\lambda > 2.5$ or $\lambda < -0.5$) these non-adjacent nodes will be removed from the Kriging configuration and the weights recalculated. This procedure is iterated until there are no big weights obtained from the Kriging solution. In Figure 11, the remaining nodes in the Kriging equation and their weights after such iterations are plotted. No more than three iterations are needed.

Figure 12 shows the histograms and maps (at the same scale from pixelplt – we can plot at “real” scale by fixing the input data) of one realization from the four situations we investigated. A cross validation exercise could be considered to test the validity of the neural network approximation.

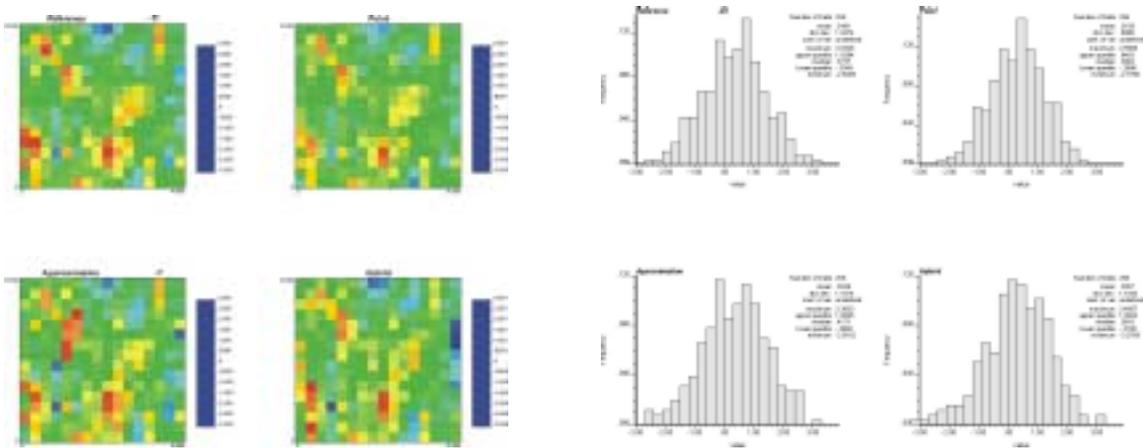


Figure 12: Histogram and map of simulated values of one realization

Future Work

We have presented some preliminary results towards the direct simulation of unstructured grids. Our effort has been on the computation bottleneck problem, that is, the fast calculation of gammabar. There are still some issues that have to be solved before practical application.

As shown at the bottom of page 2, there are three types of average variogram (covariance) involved in the Kriging equations, i.e., (1) the right hand terms that measure the closeness between the estimated block and data blocks, (2) the diagonal elements and (3) the off-diagonal elements in the left hand that are accounted for the redundancy. Their importance to the Kriging estimate and variance should be investigated. We anticipate that the off-diagonal elements are the least influential portion of the three; hence, we can take some approximations for them. The simulation result support such a hypothesis, however, care should be taken to assure Kriging weights are rational..

Neighborhood searching is another problem. The closeness of the blocks involves more than the distance between centroids. The blocks should be pre-sorted according to their centroids and size with some kind of searching tree can be constructed.

Soft data integration is another outstanding consideration. This may involve the straightforward calculation of cross gammabar (covariance) values.

Finally, the DSS algorithm that uses the Kriged mean and variance must be designed to honor the right histogram. This is the subject of ongoing research.

Acknowledgements

Chevron Petroleum Technology Company initiated this research and provided the seed funding to get it started. The funding supported Dr. YuLong Xie for six months of postdoctoral research at the Centre for Computational Geostatistics. We gratefully acknowledge Chevron's ideas, support, and openness with respect to publication.

REFERENCES

1. Aziz, K, Reservoir simulation grids: Opportunities and problems, JPT (July), 45, 1993.
2. Deutsch, C. V. and Wang, L. B., Hierarchical object-based stochastic modeling of fluvial reservoirs, Math Geology, 28(7) 857--880, 1996.
3. Ferguson, W. I. and Wadsley, A. W., The construction of curvilinear co-ordinate grid systems for reservoir simulation, SPE European Petroleum Conference, SPE 15857, October 20-22, 1986.
4. Guerlliot, D. R and Swaby, P. A., An interactive 3D mesh builder for fluid flow reservoir simulation, SPE Computer Application, 1993.
5. Heinemann, Z. and Brand, C. W., Gridding techniques in reservoir simulation, Proceedings of the 1st and 2nd International Forums on Reservoir Simulation, September 12-16 1988.
6. Heinemann, Z., Brand, C. W. and Chen, Y. M., Modeling reservoir geometry with irregular grids, SPE Reservoir Engineering, SPE 18412, 1991.
7. Journel, A. G., Past, present and future of petroleum geostatistics, SCRF, 28, 1998.
8. Pao, Y. h., Adaptive pattern recognition and neural networks, Addison-Wesley Publishing Company, Inc., 1989.
9. Pedrosa, O. A. and Aziz, K., Use of hybrid grid in reservoir simulation, 1985 Middle East Oil Technical Conference and Exhibition, March 11-14 1985.
10. Robertson, G. E., Grid-orientation effects and the use of orthogonal curvilinear coordinates in reservoir simulation, Annual Fall Technical Conference and Exhibition, SPE 6100, October 3-6 1977.
11. Rosenberg, D. W. von, Local grid refinement for finite-difference methods, SPE annual technical conference and exhibition, SPE 10974, September 26-29 1982.
12. Sharpe, H. N., A new adaptive orthogonal grid generation procedure for reservoir simulation, 1990 SPE Technical Conference and Exhibition, SPE 20744, September 23-26 1991.
13. Verma, S., Flexible grids for reservoir simulation, Department of Petroleum Engineering, Stanford University, Ph.D. Thesis, 1996.
14. Xie, Y.L. and Deutsch, C. V., Gridding in Reservoir Characterization and fluid flow simulation