

Fast SGS for Unstructured Grids

John Manchuk and Clayton V. Deutsch

Center for Computational Geostatistics
Department of Civil and Environmental Engineering
University of Alberta

This paper introduces a method of populating unstructured grids using sequential Gaussian simulation. The use of unstructured grids is becoming more popular, especially with the petroleum industry. The shift to unstructured grids is happening even before we have proven methods for populating them. Current methods for populating these grids include the following: (1) sequential Gaussian simulation on a high resolution point scale grid followed by block averaging; (2) sequential Gaussian simulation of block centroid locations at point scale; and (3) other techniques not necessarily involving kriging. The method proposed here is similar to method one, except that the refined point scale grid is generated by locally refining each unstructured volume. The result is an unstructured grid of points to be simulated and block averaged to the correct volumes. The level of refinement is block-dependent providing for very efficient simulation.

Introduction

Simulation on unstructured grids is a pressing topic in geostatistical research. Many issues arise when moving away from a regular grid including complex block geometry, calculation of average covariance values between blocks, and non-linearity of the normal score transform resulting in incorrect averages. Most current geostatistical simulation algorithms are point simulation based, which would not account for the different block volume support, or they employ average covariance calculations in Gaussian units, which retains the problem of averaging non-linearly transformed data.

The sequential Gaussian simulation algorithm (sgsim) within GSLIB (Deutsch and Journel, 1998) performs point based simulation, which is to be used for populating unstructured grids. Sgsim was extended to perform a locally refined point scale simulation within blocks of various support volumes. This method avoids the need to calculate average covariance values and also never averages information in Gaussian units. Averaging is done after back-transformation to original units which accounts for the volume of each block.

Development of the new program, called usgsim, has progressed to provide single variable simulation. Multivariate capabilities are currently being added including collocated cokriging under a Markov assumption, generalized cokriging for numerous primary variables, and the ability to simulate multiple variables simultaneously. For a more complete algorithm, other geologic features will have to be considered including facies, stratigraphy and faults.

Parameters and Program Flow

Extending sgsim to populate unstructured grids requires a specification for these grids. A new unstructured grid specification was developed to provide flexibility in defining grids for various

geologic resources. *An Interim Format and Tools to Visualize Unstructured Grids* provides an explanation of this grid specification and is available in this CCG report. For the single variable simulation version of usgsim, program flow includes two additional steps to the conventional sgsim.

1. Read in all conditioning data
2. Normal score transform the data
3. Read in the unstructured grid definition
4. Refine each grid block into a set of point-scale locations
5. Choose a random path through the point-set
6. For each point-scale location
 - a. Search for nearby conditioning and previously simulated nodes using the super-block search strategy
 - b. Setup and solve the kriging system of equations for a conditional mean, m and variance, σ
 - c. Randomly draw a uniformly distributed value, q , in $(0,1)$
 - d. Calculate y as the inverse standard normal distribution with q as the quantile: $y = G^{-1}(q)$
 - e. Assign the simulated value: $z = y \cdot \sigma + m$
7. Back-transform simulated values
8. Average the point-set and assign values to the unstructured grid blocks.

Steps that are new or different than those found in sgsim are 3, 4, 6.a and 6.b. Step 3 required a new file parsing routine. Essentially, grid components including corner-points, lines, faces and cells are input and assembled into various data structures. Cells are defined using constructive geometry. Each cell is composed of a set of faces, each face consists of a set of line segments forming a boundary, and each line segment is defined by two endpoints. This type of representation makes calculations for Step 4 possible.

Refining an unstructured grid cell is a geometric problem. A cell can be a convex or non-convex solid and it can also have curved edges (see Figure 1). In some cases, a boundary may define a face that is a cylindrical shell rather than a plane. These possibilities make cell refinement complex.

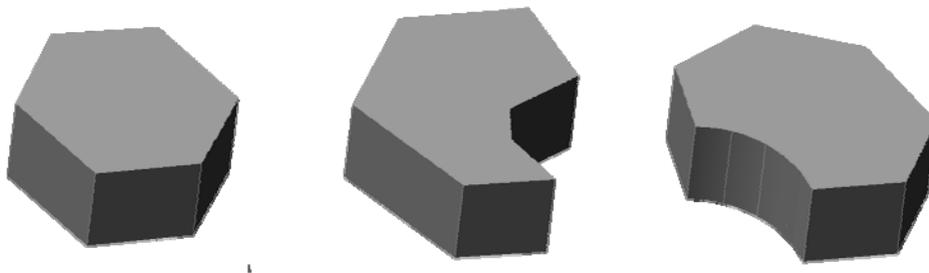


Figure 1: A convex cell (left), non-convex cell (center) and cell with a curved face (right).

For Step 4, each block is refined into a user specified number of points. Numbers of refinement points to generate in each Cartesian coordinate direction are specified as parameters. For 2 by 2 by 2 refinement, eight points would be generated. For a particular cell, the refinement process first generates an initial set of points and then adjusts those points so that they are inside the cell. A breakdown of the process is given below, after which each step will be explained.

1. Calculate the object bounding box (OBB) of a cell
2. Evenly distribute a set of refinement points throughout the OBB
3. For each point
 - a. Check that it is inside the cell
 - b. If it is not inside, redistribute points along the longest axis of the OBB
4. Save the set of points for simulation

Step 1: Calculating an object bounding box. There are two types of bounding boxes: those that are aligned to the Cartesian coordinates and those that are aligned to a coordinate system local to a particular object. In this case, the object is any unstructured grid cell. Determining the local axes is accomplished with principal component analysis (PCA). In statistics, PCA is used for explaining the variance-covariance structure of a set of variables with linear combinations of those variables (Johnson and Wichern, 2002). The linear combinations form a set of mutually orthogonal vectors: these are the local axis of highest variability for that variable set. For an unstructured grid cell the variable set comes from the x, y and z locations of the set of points defining the cell. The principal components, which are linear combinations of x, y and z, define a new orthogonal vector space for the unstructured cell. The OBB for the middle cell in Figure 1 is shown in Figure 2.

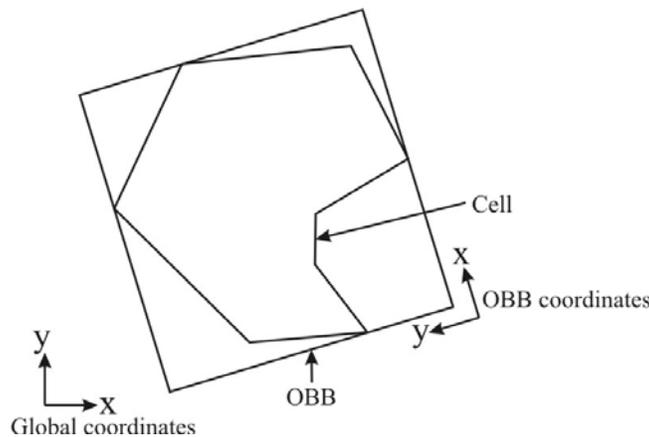


Figure 2: Top view of an unstructured grid cell and its OBB.

Step 2: Generate refinement points. An initial set of point-scale locations is defined using on-center regular grid geometry in the OBB coordinate system, which is denoted with x' , y' and z' . We know the number of points to create in each direction, n_x , n_y , n_z , the dimensions of the OBB, d_x , d_y , d_z , and the center position of the OBB, c_x , c_y , c_z . For an index i , j , k , with $0 < i, j, k \leq n_x', n_y', n_z'$, the points position, p_x , p_y , p_z , is calculated using Equation 1:

$$\mathbf{p} = \left(\mathbf{c} - \frac{(\mathbf{n} - 1) \cdot \mathbf{d}}{2 \cdot \mathbf{n}} \right) + (\mathbf{I} - 1) \cdot \mathbf{n}$$

Where

$$\begin{aligned} \mathbf{p} &= \langle p_{x'}, p_{y'}, p_{z'} \rangle, \mathbf{c} = \langle c_{x'}, c_{y'}, c_{z'} \rangle \\ \mathbf{n} &= \langle n_{x'}, n_{y'}, n_{z'} \rangle, \mathbf{d} = \langle d_{x'}, d_{y'}, d_{z'} \rangle \\ \mathbf{I} &= \langle i, j, k \rangle \end{aligned} \quad (1)$$

Even though the points are generated in the local coordinate system, they can be easily transformed to the original or global coordinate system. The transformation matrix is a symmetric matrix that stores each linear combination vector. In this case, the matrix is 3 by 3. Transforming locations from x, y, z space to x', y', z' space is done using this matrix. Back-transformation is done using its inverse. The set of points using 2 by 2 by 2 discretization in the global coordinate system is shown in Figure 3.

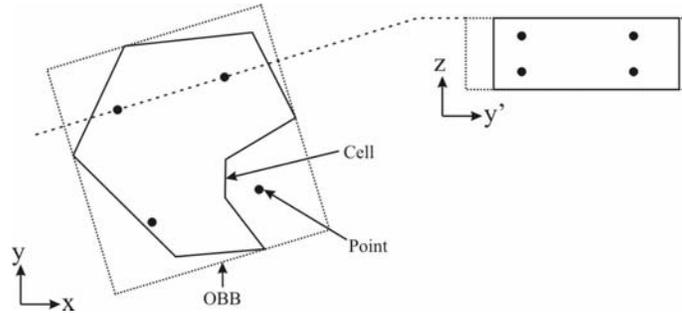


Figure 3: Initial point set refined using the OBB coordinate system.

Step 3: Checking and adjusting points. Checking if a point is inside a potentially non-convex polygon involves projections and polygon point containment (PC) tests (Mortenson, 1999). For a particular point, the first step is to calculate all intersections with the cell and the OBB axes with the point acting as the origin. PC-tests can then be done in each direction using the intersections, see Figure 4. If any one of the tests determines that a point is outside, then the point is not in the cell.

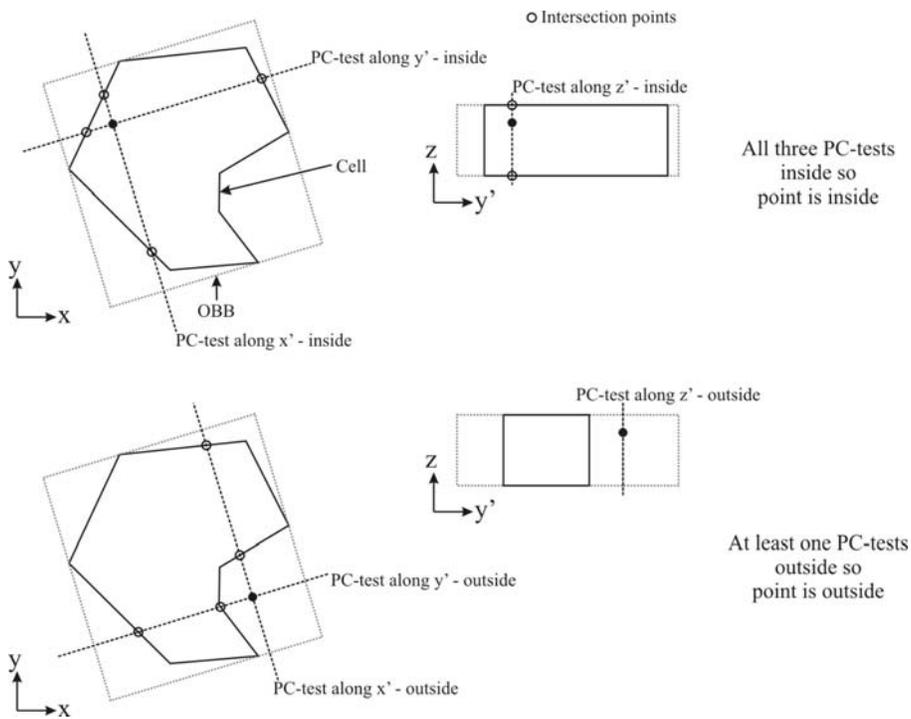


Figure 4: Point containment test for an unstructured grid cell.

Points that are not inside the cell are translated along the longest OBB axis such that they are inside. For the case where the point, p , is outside in Figure 4, we redistribute all refinement points along x' , with p as the origin, to be between the intersection points (see Figure 5). The process of generating point sets when considering curved faces is similar; the difference is in calculating intersection points along the OBB axes.

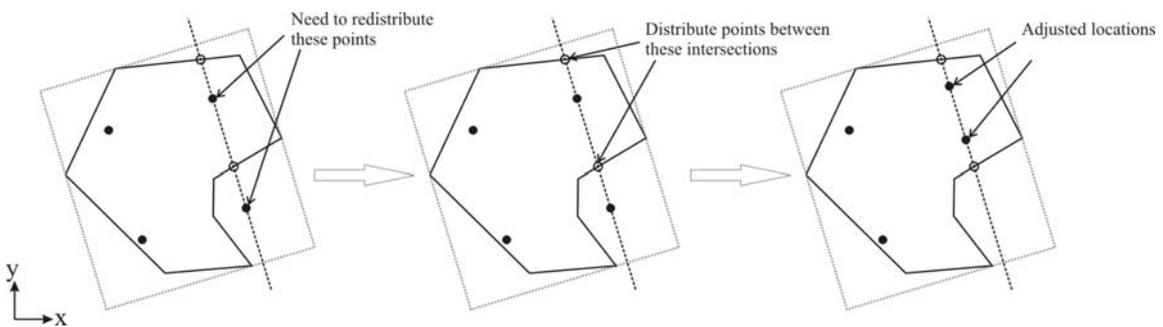


Figure 5: Point adjustment process for points that are outside a cell.

It was mentioned that the search for nearby conditioning data was different than that for *sgsim* (Step 6.a. from the above list). A super block search is still utilized; however, we are no longer searching for previously simulated nodes that fall in a regular pattern. The refined point set has no structure. We must search for previously simulated points the same way as with irregularly spaced conditioning data. This slows program execution to some degree. Execution time is again slowed by the difference in Step 6.b. With regular grids, an efficient covariance lookup table can be created. The distances and orientations of grid nodes relative to one another are constant.

This way, covariance can be looked up based on indices. With irregularly spaced point data, covariance values and distances are calculated directly for every pair in the kriging system.

Adequate Refinement

Determining a reasonable number of refinement points for a grid cell can be done with a single grid block and some synthetic data, see Figure 6. Refinement points were slowly increased and for each number, 100 realizations were generated for the blocks value. Means and standard deviations of the realizations were calculated for each different set of refinement points. Fluctuations in these means and standard deviations were compared with the number of refinement points used, see Figure 7. Fluctuations level off with approximately 20 refinement points.

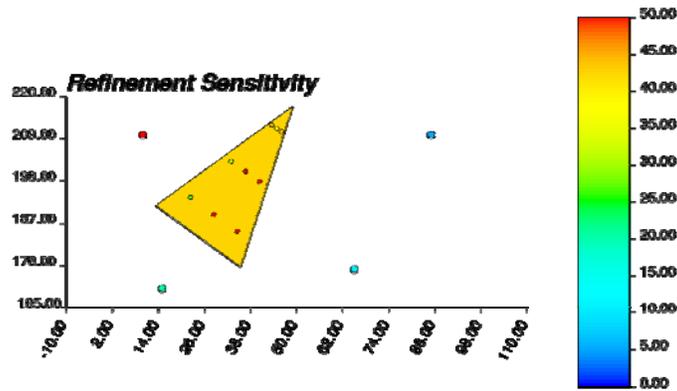


Figure 6: A simple block and synthetic conditioning data (large bullets). A set of refinement points and their simulated values is also shown (small bullets).

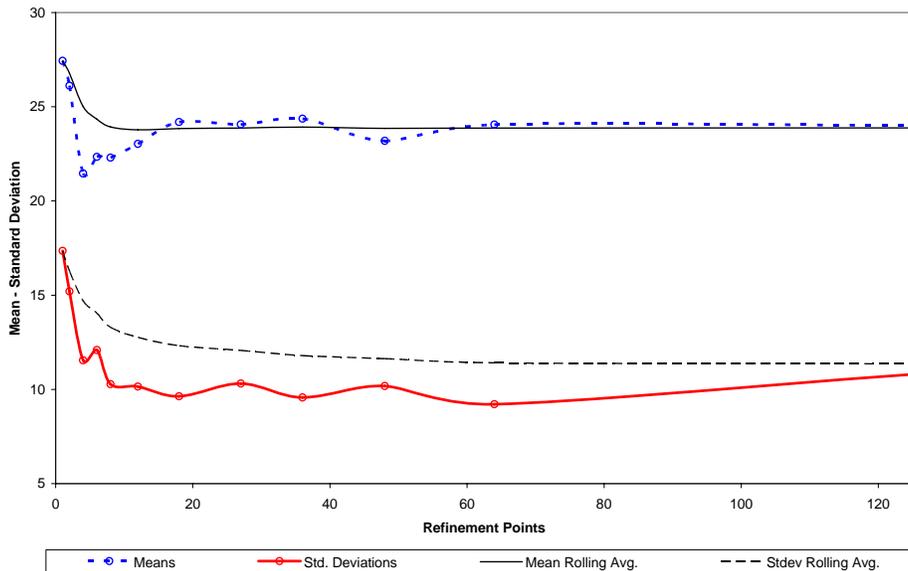


Figure 7: Mean and standard deviation fluctuations with number of refinement points.

Program Testing

Testing of usgsim will be done with a small unstructured grid of 35 aerial cells. The grid has one cell in the z-direction. Thickness of cells was set at 10 units. A synthetic data set was created as well with 18 data points. One of the tests that will be carried out involves performing sequential Gaussian simulation (SGS) on a regular grid and block averaging results to the unstructured grid cells. Conditional mean and variance values of each block can be checked to ensure usgsim is producing valid results. A CPU time comparison can also be done. This will compare the time it takes to simulate on a fine-scale regular grid and block average to the unstructured grid versus the time to run usgsim with the unstructured grid directly.

Establishing a reasonable regular grid depends on the smallest cell in the unstructured grid and the number of discretization points to be assigned to each cell. The resolution of the regular grid should be high enough to place that approximate number of discretized points in the smallest unstructured cell. The sample unstructured grid layout and synthetic data are shown in Figure 8.

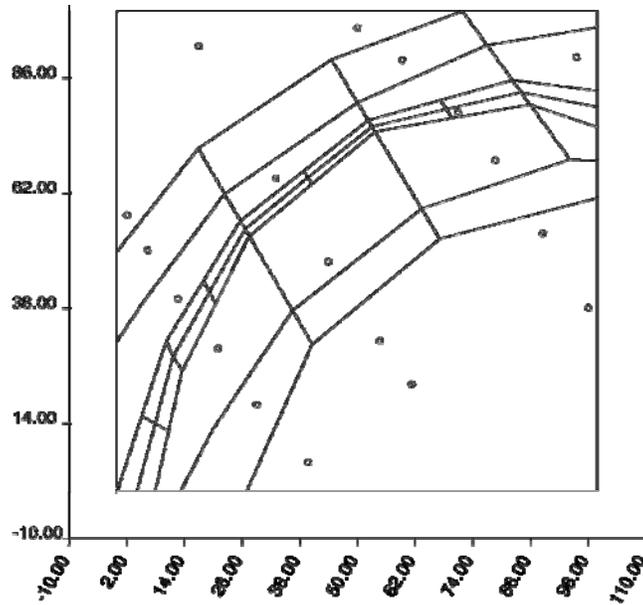


Figure 8: Sample unstructured grid layout and synthetic data.

To keep calculations simple and since the unstructured grid is only one cell thick, cells will be discretized in the z-plane only. Regular grids will be kept in two dimensions. Looking at the smallest cell in the grid, a relation can be defined between number of refinement points in the cell and required regular grid resolution. Area of this cell, A , is 23.9502 or approximately 24 units². Since the overall grid is square, regular grid cells will be square as well with side length d . If the number of points to be placed in the smallest unstructured cell is n , the approximate regular grid cell size can be defined by Equation 2.

$$d = \sqrt{\frac{A}{n}} \quad (2)$$

Since we also know the origin and dimensions of the overall unstructured grid (0,0 and 100.0,100.0) the regular grid definition can be generated. A table of number of refinement points and regular grid resolution has been put together in Table 1.

Table 1: Regular grid resolution for given refinement points.

Refinement Points, n	Side length, d	number of blocks in each direction	Adjusted side length	regular grid origin	total cells
2	3.46	29	3.45	1.72	841
4	2.45	41	2.44	1.22	1681
8	1.73	58	1.72	0.86	3364
16	1.22	82	1.22	0.61	6724
32	0.87	115	0.87	0.43	13225

To make sure that the smallest unstructured cell always has some regular grid blocks within (based on center point geometry) the number of refinement points will be kept high enough such that the side length, d, is at most the smallest dimension of the unstructured cell. This length is 1.23 units. There must be at least 16 (4 by 4) refinement points. Figure 9 shows two realizations; one simulated with usgsim and 16 refinement points and one simulated with sgsim and block averaged. Note that the random path through nodes is different for each method; results will not look similar for the same random number seed. Expected mean and standard deviation values of each block can be compared over a set of realizations however.

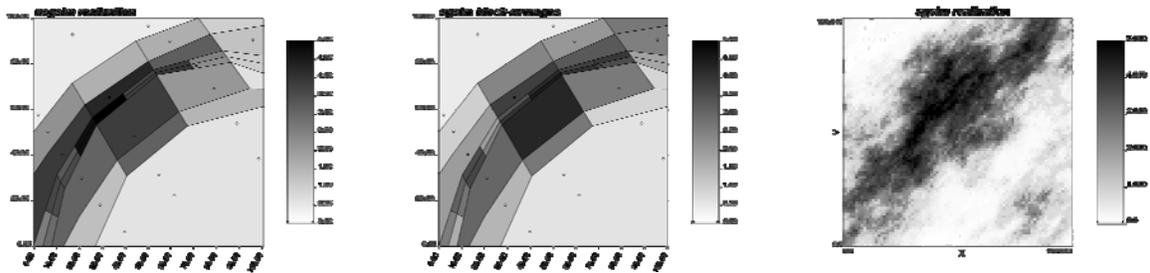


Figure 9: Simulations using both usgsim and sgsim.

Figure 10 shows a comparison of the expected mean and standard deviation values of each block for 100 realizations calculated from usgsim and from block averaged sgsim runs. A slight bias can be seen in both plots as the points fall below the line. This implies that the mean and standard deviation is higher for usgsim.

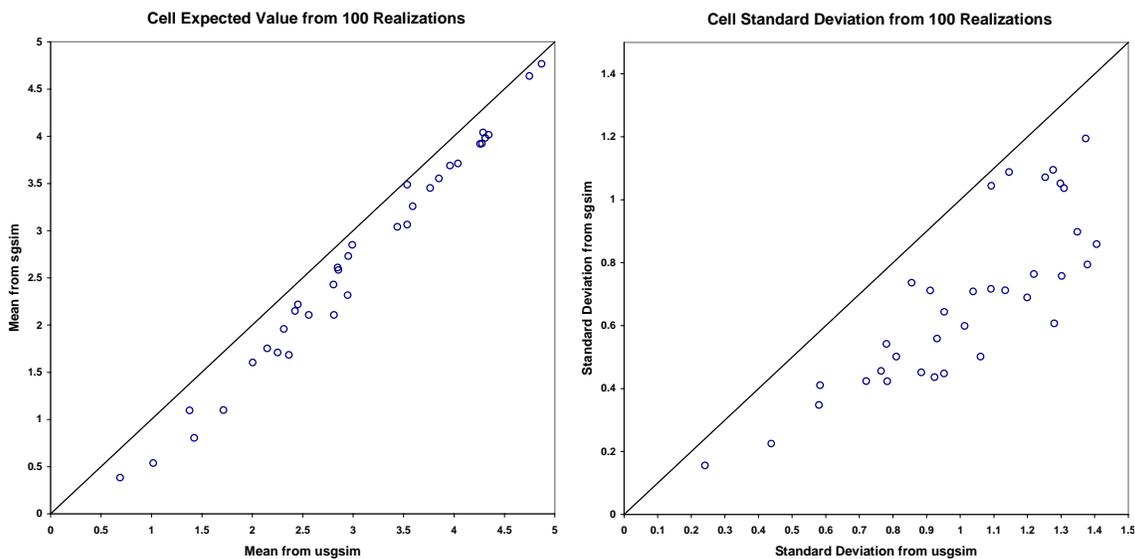


Figure 10: E-type mean (left) and standard deviation (right) comparison charts for usgsim and sgsim.

Another comparison is regarding memory. When using 16 refinement points, the total file size for usgsim, which output simulated values at refinement point locations and for grid blocks, is 2.3 MB whereas for sgsim the file size is 10.5 MB. Keep in mind this is only for a 35 cell unstructured grid. For this particular grid setup, runtime for both programs was identical. Runtimes for sgsim would increase beyond those of usgsim if the smallest unstructured cell were $\frac{1}{2}$ or a $\frac{1}{4}$ the size of that used. In this case, regular grid resolution would be 1.41 or 2 times finer than indicated in Table 1.

Model Checking

Some of the standard procedures for validating models generated with sequential Gaussian simulation include histogram reproduction and variogram reproduction. These are straightforward for regular grids. The area of interest is simulated at equally spaced intervals with identical volume support. Even if the support volume is different than that of the input data, the histogram and variogram can be scaled appropriately using affine correction or the discrete Gaussian model. With unstructured grids, the support volume of cells is not constant. Histogram reproduction cannot be checked easily. Checking the histogram with the refined set of points is not correct either. The points are not distributed evenly over the area of interest. Declustering of the refined point set may be one possible way to check histogram reproduction.

The same problems persist for variogram reproduction. Variograms would have to be calculated over the refined point set. Calculating variograms over irregularly spaced data can be very time consuming, especially when the refined point set becomes vary large, possibly in the millions for large grids.

Future Work

More work is planned for improving the algorithm for Gaussian simulation on unstructured grids. Currently, only one variable can be simulated. This is going to be expanded to cover multivariate

simulation. Multiple input variables will be accepted and any number of them can be simulated. Multiple inputs will be available for non-located data, in which case a linear model of coregionalization must be provided, or for collocated variables under a Markov assumption (Chiles and Delfiner, 1999). Simulation of multiple variables is accomplished with simultaneous kriging where multiple kriging weight vectors are solved for in a system of linear equations involving multiple right-hand-side vectors.

Work regarding incorporation of other geologic features such as stratigraphy and faults into simulation is also needed. Performing a stratigraphic coordinate transform is typically carried out prior to simulation. This removes any curvilinear features in the stratigraphic layers, which simplifies variogram calculation and searching for conditioning data, see Figure 11. Another aspect regarding stratigraphy is identifying layers that are geologically different. Data from these layers may be considered different for kriging. Faults can be used in a similar fashion for coordinate transformation. Any topological changes that result from faulting are undone by the transform, see Figure 11.

Generating unstructured grids prior to removing stratigraphic and fault features may result in cells that intersect layers and/or faults. These blocks cannot be identified as belonging to a specific layer. Upon being refined, some points may exist on opposite sides of a fault or of a stratigraphic surface. Simulating and block averaging these points will likely result in undesirable outcomes.

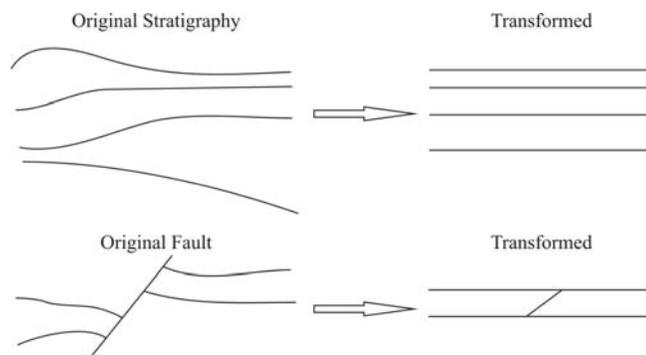


Figure 11: Transformations to remove stratigraphic and fault features.

Another issue is simulation of categorical variables such as facies types. For each block in a regular grid, kriging provides probabilities for each facies type. These can be constructed into a cumulative distribution from which a facies type can be simulated. For an unstructured grid this is not so straightforward. For a set of refinement points in an unstructured cell, kriging and simulation will give the same information. To assign a single facies type to the unstructured cell requires combining the facies probability distributions from each refinement point, then drawing a value. Combining distributions is another component of future work. Since simulation is carried out on refinement points only, each point in a cell must be assigned a simulated facies type. For a particular cell, probability distributions at each of its refinement points must be saved during simulation until all points are simulated. Depending on the random path through which points are visited as simulation progresses and the number of facies categories this would require more memory than a continuous variable.

Other future work involves making improvements to cell refinement. Some degenerate cases for determining if points are inside a non-convex solid exist. These need to be understood and included in the usgsim program. These problems may be contributing to the difference in runs between usgsim and sgsim identified in Figure 10. One issue is that points are not evenly

distributed throughout a particular unstructured cell. Ideally, we would like a set of refinement points to divide a cell into equal volumes. This is a complex problem that would likely require too much time to execute for a large number of unstructured cells. However, the current method should be improved to offer better refinement.

Conclusions

This work has shown that sequential Gaussian simulation can be performed on unstructured grids in a more elegant way than just block averaging a high resolution point scale regular grid. Results are correct to the accuracy of block discretization. Depending on the unstructured grid definition, usgsim may perform faster than sgsim and block averaging. This is true for grids with a high ratio of largest to smallest cell volume. Sgsim is more demanding in terms of memory as well. Validating models is a disadvantage for unstructured grids. Histogram reproduction cannot be confirmed. Variogram reproduction can become a time consuming exercise for large grids.

References

- Deutsch, C.V. and Journel, A.G., *GSLIB Geostatistical Software Library and User's Guide*, Oxford University Press, 1998
- Chiles, J. and Delfiner, P., *Geostatistics, Modeling Spatial Uncertainty*, John Wiley & Sons Inc., 1999
- Johnson R.A. and Wichern D.W., *Applied Multivariate Statistical Analysis*, Prentice Hall, 2002
- Mortenson M.E., *Mathematics for Computer Graphics Applications*, Industrial Press Inc., 1999