

A Second Generation Gibbs Sampler for Simulation with Multiple-Point Statistics

Steve Lyster

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

Simulation of categorical variables with multiple point statistics (MPS) is being used increasingly for the construction of numerical models that respect a variety of data including curvilinear features. The challenges of MPS include (1) identification of an appropriate training image, and (2) development of flexible algorithms to impose the statistics on simulated realizations. This paper presents improvements to a Gibbs Sampler-based algorithm to impose MPS on simulated realizations. The implementation details necessary to create viable large-scale realizations are addressed. The algorithm is ready for field testing.

Introduction

Multiple-point statistics (MPS) methods are used to represent geologic features in more realistic and robust ways than traditional geostatistics. Rather than considering linear estimates based on individual conditioning points, MPS use several points simultaneously to determine conditional probabilities. These higher-order relations describe complex features beyond the standard Gaussian model of spatial structure.

Several methods for reproducing MPS have been considered: the single normal equation (SNE) approach first proposed by Guardiano and Srivastava (1993) and further expanded by Strebelle and Journel (2000, 2001) and Strebelle (20002); simulated annealing (Deutsch, 1992; Lyster et al, 2004a, 2004b); Gibbs sampling (Srivastava, 1992, Lyster et al 2006); a neural network iterative scheme (Caers and Journel, 1998; Caers, 2001); and integration of runs with indicator simulation for reproduction of higher-order relations in continuous data (Ortiz, 2003).

MPS methods have primarily focused on categorical indicators and modeling rock types; this is probably due to the relative ease with which training images may be created for geologic structures, as opposed to the dense sampling necessary to infer high-order moments for continuous data. Within the family of categorical methods, most use iterative approaches to reproduce MPS. This is due to the reduction in dimensionality of the statistics that must be calculated and stored when using a template.

The approach proposed in this paper is similar to several of the above-mentioned methods in that it is an iterative simulation algorithm meant to be used for modeling of geologic facies or rock types. While a Gibbs sampler method has been investigated before, a novel technique for determining the conditional probabilities will be presented. This new technique improves both reproduction of features and time requirements for simulation compared with previous implementations of the Gibbs sampler. Two examples will be shown of conditional simulations and the results analyzed briefly.

Gibbs Sampler

The Gibbs sampler is a statistical algorithm that was first proposed by Geman and Geman (1984); it is a special case of the Metropolis algorithm (Metropolis et al, 1953). It is a method used for drawing samples from complicated joint or marginal distributions, without requiring the density functions of the distributions (Casella and George, 1992). In the context of geological modeling, a sample from the joint distribution is a simulated realization. In a Gibbs sampler only the conditional distributions are required to sample from the joint distribution, which is why this approach is attractive as a MPS simulation method; the conditional probabilities are determined from a training image.

The basic idea of a Gibbs sampler is that of resampling individual variables conditional to others in the same sample space. Drawing new values conditional to all others, and repeating this process many times, results in an approximation of the joint (and marginal) distribution(s). For example, to determine the

density function for $f(X,Y,Z)$, one would start with an initial state (values) for X , Y , and Z and then sequentially draw values from the conditional distributions $f(X|Y,Z)$, $f(Y|X,Z)$, and $f(Z|X,Y)$. When a number of conditional values have been drawn, the state of (X,Y,Z) approximates a sample from the joint distribution $f(X,Y,Z)$. Repeating this process enough times allows the density (or histogram) of the joint distribution to be constructed empirically. In geostatistics generating many samples from the joint distribution is analogous to creating many simulated realizations.

The Gibbs sampler is a Markov Chain Monte Carlo (MCMC) method (Gelfand and Smith, 1990), meaning that each new value drawn from a conditional distribution is assumed to be dependent only on the current state of the variables. While theoretically the new value should be dependent on all prior states, because the current state is dependent on the previous state and the previous dependent on the one before that; thus, using only the current state to determine the new value drawn accounts for all earlier states and therefore the theory may be simplified to ignore all past states and use only the current. This property is known as conditional independence, and greatly simplifies the use of MCMC methods.

An advantage of the Gibbs sampler workflow for geospatial modeling is that there is no strict specification for how the conditional distributions need to be determined. Previous use of the Gibbs sampler (Srivastava, 1992) focused primarily on using kriging and two-point statistics; a single normal equation using MPS was also explored briefly. Besides a MPS template, it would be possible to integrate a variety of statistics into the conditionals, such as locally varying means, secondary data, and lower-order statistics to ensure their proper reproduction.

Estimation of Conditional Probabilities

The “true” conditional probabilities are not available for use in this case of the Gibbs sampler, so they must be estimated. To find the conditional probabilities of facies at a point, any desired method could be used. However, the quality of the resulting realizations will be determined by the conditional probabilities selected. For example, using only the univariate histogram as a conditional distribution would guarantee that the facies proportions are matched exactly, but would not give any structure in the results; using the variograms and kriging to determine conditional probabilities would reproduce the direct covariance structure but nothing more; adding the cross-variograms in a licit LMC would reproduce the relations between facies as well. MPS reproduces higher-order statistics in the results.

The most basic method of using MPS in the conditional distributions for a Gibbs sampler would be to use Bayes’ Law to directly infer the probabilities of each facies from the training image. An example of a problem in using this approach is illustrated in Figure 1. The central point in the given MPS template, and therefore the point for which the conditional probabilities are needed, is marked with an “X”. The arrangement of black and white facies in the 16-point template shown may not appear in the TI; with 16 points, there are $2^{16} > 65,000$ combinations, not all of which will be informed.

The standard procedure would be to drop the farthest away point from the central location, and recalculate the conditional distributions. However, if too many points are dropped there may not be enough high-order structure contained in the conditional distribution, leading to poor quality realizations. Iterative simulation algorithms will encounter this problem many times early in the simulation process as the image will not yet contain much high-order structure and the estimated conditional probabilities are therefore likely to get “stuck”, requiring a better image for better estimates but requiring better estimates to improve the image.

Dividing the template into smaller sub-templates increases the proportion of informed classes and helps alleviate the problem of inference. The template is divided into discrete multiple-point events (MPEs) that are used as information for estimating the conditional probabilities. Figure 2 shows the same information as Figure 1, but divided into four MPEs of four points each. A four-point MPE has only $2^4 = 16$ possible classes, so a fully informed *multiple-point histogram* (Deutsch, 1992) is much easier to characterize. Using this idea, linear weights may be used to estimate the conditional probability of each facies:

$$P^*(k) = \sum_{i,\alpha} \lambda_{i,\alpha}^k \left[I(E_i^\alpha) - P(E_i^\alpha) \right] + P(k) \quad (1)$$

where $P^*(k)$ is the estimated probability of facies k at the current point; $\lambda_{i,\alpha}^k$ is the weight given to the MP histogram class α of MPE i for the probability of k ; E_i^α is class α of MPE i ; $I(E_i^\alpha)$ is the indicator of class α of MPE i , which is 1 for those MPE classes which are present and 0 for those that are absent; $P(E_i^\alpha)$ is the global probability of E_i^α ; and $P(k)$ is the global probability of k , also denoted as P_k . This approach takes into account not only those MPE classes which are present near the point to be estimated, but those that do *not* occur, which in some cases is even more important.

The optimal estimation weights $\lambda_{i,\alpha}^k$ are found using a set of kriging-like equations determined by minimizing the estimation variance:

$$\sum_{j,\beta} \lambda_{j,\beta}^k \cdot Cov\{E_i^\alpha, E_j^\beta\} = Cov\{E_i^\alpha, k\} \quad (2)$$

This system of equations requires the *multiple-point covariance*, defined as

$$\begin{aligned} Cov\{E_i^\alpha, E_j^\beta\} &= E\{I(E_i^\alpha) \cdot I(E_j^\beta)\} - E\{I(E_i^\alpha)\} \cdot E\{I(E_j^\beta)\} \\ &= P(E_i^\alpha \cap E_j^\beta) - P(E_i^\alpha) \cdot P(E_j^\beta) \end{aligned} \quad (3)$$

where $Cov\{E_i^\alpha, E_j^\beta\}$ is the covariance between $I(E_i^\alpha)$ and $I(E_j^\beta)$, calculated from the TI.

The previous implementation of this approach (Lyster et al, 2006) used only those MPEs which are present to estimate the conditional probabilities, solving the system in Equation 2 at each location. While this methodology did show some promise, improvements to both the estimation variance and algorithm speed were obtained by using the full set of MPE classes, even those with indicator zero. The system is only solved once in this case, and the weights are stored for later use. The system is singular due to the exhaustive and exclusive properties of indicators, i.e. every MPE at each location has a single class which occurs and has indicator one, and all of the rest of the indicators for that MPE are zero. A singular system of equations has an infinite number of solutions and a single one must be selected for use as the estimation weights. The system in Equation 2 is solved in this case using eigenvalue decomposition and singular value decomposition to find the set of weights which have the lowest norm while having the optimal error variance. This is accomplished through the use of the `dsyevd` routine in the LAPACK (Linear Algebra PACKage) suite of tools (Anderson et al, 1999).

“Base” estimates may be found with the optimal weights using the equation:

$$P^+(k) = -\sum_{i,\alpha} \lambda_{i,\alpha}^k \cdot P(E_i^\alpha) + P(k) \quad (4)$$

Then, at each location visited in the simulation path, the estimates are calculated from the base estimates, estimation weights, and MPE indicators:

$$P^*(k) = P^+(k) + \sum_{i,\alpha} \lambda_{i,\alpha}^k \cdot I(E_i^\alpha) \quad (5)$$

Using this approach gives the lowest possible error variance for the estimated probabilities of each facies, for a defined template of MPEs. To compare the estimator in Equation 1 with the approach from last year (Lyster et al, 2006), consider the training image shown

Using the TI in Figure 3 to calculate the estimation weights and the MPEs shown in Figure 3, the estimates of the three different facies are shown in Table 1, it can be seen that there is significant reduction in the error variances, as well as additional stability in the estimates as the cpdf value of facies 3 is very near zero rather than being a large negative. Other similar comparisons yielded comparable results.

Modifications to Conditional Probabilities

Once the conditional probabilities have been determined, some modifications are necessary to correct undesirable results from the methodology and improve reproduction of the TI. The first correction to be considered is the simplest: those estimated cpdf values which are negative are reset to zero. This is a standard implementation of categorical simulation (Deutsch and Journel, 1998).

The next correction is to “steer” the simulation toward the correct univariate facies proportions. This is called a *servosystem* correction (Strebelle and Journel, 2001). The difference between the target global pdf, $P(k)$, and the current realization pdf, $P^C(k)$, is multiplied by a correction factor and added to the cpdf :

$$P'(k) = P^*(k) + \mu \cdot (P(k) - P^C(k)) \quad (6)$$

The parameter μ , which is bounded by zero (no correction) and one, is used to allow the user to set the level of servosystem. Once the servosystem correction has been added, a *noise reduction factor* is used to remove excess randomness which tends to appear over the course of several iterations. This factor, NRF , is applied by setting any cpdf value to zero if the following relation holds:

$$P'(k) < P(k) \cdot NRF \quad (7)$$

So if the estimated conditional probability of any facies, after the servosystem correction, is less than a defined fraction of its global pdf, it is set to zero. This removes any facies with small but positive probabilities of occurring, such as sand in the middle of a shale layer or waste in the middle of an ore vein. Like the servosystem parameter μ , NRF is bounded by zero (no correction) and one.

The next modification to the conditional probabilities is an adjustment to compensate for differences between the target global proportions and the proportions calculated from the TI. A Bayesian updating step is used for this (Liu, 2006). The modification is as follows:

$$P''(k) = \frac{P^G(k) \cdot [1 - P^{TI}(k)] \cdot P'(k)}{[P^G(k) - P^{TI}(k)] \cdot P'(k) + P^{TI}(k) \cdot [1 - P^G(k)]} \quad (8)$$

where $P'(k)$ is the conditional probability as modified from Equations 6 and 7, $P^G(k)$ is the global target proportion of facies k , and $P^{TI}(k)$ is the TI proportion. This relation retains rank order of the estimated probabilities and does not change those conditional proportions which are equal to zero or one.

Once the global target proportions have been taken into account, the local distributions accounting for seismic, vertical proportions or other soft information are matched by using the assumption of independence. The modified estimates from Equation 8 are updated in this way:

$$P'''(k) = P''(k) \cdot \left(\frac{P^L(k)}{P^G(k)} \right)^Q \quad (9)$$

where $P'''(k)$ is the conditional probability from Equation 8, $P^L(k)$ is the target local proportion, $P^G(k)$ is the target global proportion, and Q is the quality of the locally varying proportions. In cases where local proportions are not used, the local distributions are automatically set the same as the global pdf. An advantage of using Equation 9 to honour local proportions is that facies that have local values of zero will have their conditional distributions set to zero, and will not appear in regions where they should not.

New Simulation Algorithm

The algorithm proposed to use the concepts discussed above follows the overall workflow:

1. *Read in the training image and MPS template.* If no template exists, one is automatically created using the method proposed in Lyster and Deutsch, 2006. Once the template has been created, the algorithm stops to allow the user to review the template before proceeding.
2. *Calculate the statistics from the TI and solve the system in Equation 2.* The full system is solved, all weights are calculated, and the results are output to a file. If the file already exists these first

two steps are skipped and the weights are read in immediately. A cutoff may be specified for a minimum frequency of classes within MPEs, below which a class will not be included and will be assigned a weight of zero. This is mainly to remove those classes which are uninformed, but may be used to trim the size of the problem by not including very infrequent classes.

3. *Prepare for the simulation.* This step includes reading in conditioning data, expanding the grid to account for edge effects (see below), assigning local pdfs to locations, calculation of the base estimates as in Equation 4, allocating arrays, and preparing the output file.
4. *Assign conditioning data to the nearest node on the current grid.* The conditioning data are assigned as grid nodes and locked in place.
5. *Create a random path on the current grid and create an initial image.* The random path visits all uninformed data locations on the grid in a random order. This is accomplished by assigning each location a random number and sorting from lowest value to highest. While each location is being assigned a path value, it is also assigned a random initial facies.
6. *Visit each location and resample.* The path from Step 5 is used and the conditional probabilities of all facies are calculated. Values are drawn at every location from these conditional pdfs. This step is repeated for all locations as many times as necessary until the stopping criteria are met.
7. *Check the stopping criteria.* Continue on to Step 8; otherwise, repeat Step 6.
8. *Move on to the next grid.* The conditioning data are unfrozen, the simulated values on the current grid are frozen, and the algorithm continues from Step 4 on the next finest grid.
9. *Write to file and continue.* If the number of desired realizations has been simulated, the algorithm is done. Otherwise, repeat from Step 4 on a new, empty model.

Stopping Criteria

All iterative simulation methods require stopping criteria. CPU time must be balanced with improvements in the realizations. In some cases, running the simulation too long may lead to artifacts (Lyster and Deutsch, 2007). There are numerous stopping criteria. The method used previously (Lyster et al, 2006) was simply to loop over all unsampled locations a defined number of times and then stop. This number was typically found to be about ten, but could vary. Simply using a set number is not the most efficient way to obtain the best possible realizations for the least amount of CPU time.

The first, and primary, stopping criterion proposed to be used in the algorithm involves tracking the number of values which are changed in each loop over all locations, and stopping when this number approximately levels off. An example of this is shown in Figure 4. The number of changes at each loop for a simulation is graphed, along with the correlation coefficients for the last five loops. With a random initial image the number of changes is initially very high but as features and relations are introduced the changes are quickly reduced. The correlation coefficient of the last five loops is used to identify when the number of changes levels off and reaches a stable value. When the correlation coefficient first becomes non-negative, the slope of the graph of number of changes is no longer negative and the number has become approximately stable. The first occurrence of a correlation greater than or equal to zero is the signal to stop the simulation and move on to the next grid or realization.

In Figure 4 it may be seen that after an initial drastic drop in the number of changes for each loop, there is a long gradual decline. At some point random chance will result in this gradual decline giving a correlation greater than zero and the simulation is stopped. However, in a few select cases the number of changes may have too negative of a slope and even random noise will not result in the first stopping criterion being met. For this reason, there are two secondary criteria. The first of these is a threshold for the number of changes: if fewer than, say, 0.1% of all locations are being changed in a loop then the simulation is not changing much and further loops are only wasting time and resources.

The other secondary stopping criterion is a maximum number of loops. This prevents the algorithm from becoming trapped in an infinite loop. This number is typically set at 50.

Multiple Grid Implementation

In order to better reproduce long-range features present in the TI, multiple grids are used. This is accomplished by simulating every n th node first, then freezing these simulated values in place and using them as conditioning for the next grid. As currently implemented, the grids work on powers of two; so, if four grids are to be used the first will simulate every 8th node, the second every 4th, the third every other node and the final grid will always simulate every location in the desired grid. Figure 5 shows a small example of two grids, one at 4x4 discretization and the other at 2x2.

Honouring of Conditioning Data

As an iterative method, the Gibbs sampler may result in artifacts near hard conditioning data. To remedy this the hard data points are honoured at each of the multiple grids; for the first grid there are typically enough conditioning data to enforce the proper structure in the relatively few nodes being simulated, and the facies which are in relation to the conditioning are in the proper locations as the simulation proceeds. In order to properly honour the conditioning data, the nearest grid locations at the current discretization are assigned the informed values before the resampling portion of the algorithms progresses. Once the grid is finished, the conditioning data are unassigned from their current locations and reassigned to the nearest cell at the next finest grid. The former conditioning points are then not frozen, and are resampled in the next grid. This prevents non-conditioning points from being frozen as a single value in every realization.

Edge Effects

The iterative method proposed here uses the assumption that every surrounding cell is informed to improve computational efficiency and reduce CPU time. Near the edges of the simulation grids, this assumption does not hold. In order to ensure every point is informed, even near the edge, there are several possible solutions. The initial random values near the edge could simply be retained; the grid could be wrapped to the other side; the grid could be reflected back on itself; or alternative conditional distributions could be used. All of these options except the last will cause artifacts, and calculating alternative conditional pdfs takes extra time that may offset the speed advantages.

In order to minimize the impact of edge effects, additional cells are added on all sides of the model and then trimmed after the simulation is complete. This effectively moves the artifacts to regions which are not retained. Currently the grid is simply reflected back onto itself to inform every location. The reasons for this are: retaining pure random noise could result in artifacting in the same way as reflection (propagation beyond the buffer zone), but additional continuity is preferable in this case to reduced spatial structure; the artifacts from wrapping the grid are much longer-range and could be apparent even with a “buffer” zone added on all sides of the grid; and additional exact cpdf calculations near the edge of the grid are expensive computationally and would reduce a major advantage of the Gibbs sampler algorithm, which is speed.

In addition to the extra grid nodes, to reduce the proliferation of edge effects only those grid nodes which are in the original grid are frozen after each grid discretization is simulated; those in the buffer zone are assigned new random values at the next grid. This erases all artifacts at the conclusion of each grid and reduces the likelihood of a long-range edge effect appearing in the final results.

Example – Fluvial Channels, Two Facies

Figure 7 shows an image of fluvial channels characteristic of many petroleum reservoirs. The image was created using the FLUVSIM program and then used as the “true” data for this example. The field under study is 100 pixels square, with two facies: background (white, code zero) and channels (black, code one). Also shown in Figure 7 is a location map of thirty samples taken from the data. The proportion of channel facies is 30% in the true data and 33.3% in the sample data.

The training image used for simulation in the example is shown in Figure 8. The TI is 200x100 cells, which allows better statistical inference in the direction of greatest range of continuity. The facies proportions are the same in the TI as in the true data, 30% channel (black) and 70% background (white).

For the example 100 realizations were produced using the new proposed version of the Gibbs sampler algorithm. The template used for simulation had 14 four-point MPEs (for a total of 56 points), four grid discretizations were used for the simulation, the servosystem parameter was 0.8 and the noise reduction

factor was 0.3. All realizations converged to a solution by using the first stopping criterion, a positive correlation over the number of changes in the last five loops.

Solving the system of equations and calculating the optimal weights took 0:07 (seven seconds) on a Dell Precision 670 with dual 3.2 GHz processors and 3 GB of RAM, and the simulation of 100 realizations took 1:55. Note that although the machine used had dual processors the algorithm is not currently set up to take advantage of parallel processing. Four of the realizations are shown in Figure 9. From Figure 9 it can be seen that the overall channel structure of the true data and TI is reproduced. There is some curvilinearity present and most realizations have about 1-2 channels which span the entire width of the field, compared to the six channels in both the true and training images. Long-range connectivity has been an issue with other simulation methods, including those using MPS (Strebelle, 2002). There do not appear to be any edge effects due to the Gibbs sampler algorithm. There is some noise that seems to be caused by the multiple gridding, characterized by the checkerboard patterns seen in some regions; this may warrant further exploration into the multiple gridding approach. The global proportion of 30% channel was matched, at a 32.9% average over all realizations. This is close to the proportions contained in the sample data of 33.3%.

The probability of channel facies was calculated over the 100 realizations, and the results are shown in Figure 10. It may be easily seen that the conditioning data are explicitly honoured and appear as black and white spots on the map. Near conditioning data there are patches of channel or background facies, which is appropriate as there is a better chance of sampling the middle of a region than hitting the very edge of it. Some channel-like features may be seen in Figure 10; to the east and west of the ten black conditioning points there are extended areas with high probability of having simulated channel facies.

Even though the goal of the proposed Gibbs sampler algorithm is reproduction of MPS, two-point statistics are still important as they are used extensively in geostatistical modeling. Figure 11 shows the indicator variograms for the sample data, TI, true data, and ten realizations. In the X (east-west) direction, the high level of continuity present in the FLUVSIM output is not fully reproduced; however, the variograms of the simulated realizations do match the sample data variogram reasonably well. The range of the major structure east-west seems to be correct (about 25-30 cells); however, the true data and TI variograms do not reach the sill. In the Y (north-south) direction, the simulated realizations do reproduce the cyclicity seen in both the TI and true data. The sample data also show this cyclic feature, although it would be difficult to ascertain from sample data.

As a last measure of the robustness of the simulation, an accuracy plot was produced. While this would be unavailable in an actual study it is a useful tool for demonstration purposes. Figure 12 shows an accuracy plot, with the predicted proportion of facies 1 (as in Figure 10) shown on the X axis, and the actual probability of facies 1 being present at those locations on the Y axis. Lines representing a 1:1 slope and 15% error bands are also shown. All of the points are well within the 15% error bands, and are quite close to the 1:1 slope line. Taken together with Figure 10, which shows there are a number of local averages quite far away from the global proportions, the simulation seems to be quite robust.

Overall the proposed Gibbs sampler algorithm appears to work well for the fluvial channel setting shown here. The simple statistics are reproduced within a typical tolerance, the channel structures appear in the realizations with some curvilinearity and sinuosity, and the hard data are correctly reproduced. Exploration of the parameters such as the noise reduction factor, servosystem factor, and MPS template could improve the simulation results further.

Example – Eolian Sandstone, Three Facies

As a second example, an eolian sandstone setting is considered. This example is more complex than the previous one; there are three facies to consider, the structures are not all as similar, and the true data and training image are both real scanned rock samples taken from different (but similar) sources (Deutsch, 1992). Figure 13 shows the true data, which is 164 by 85 pixels in size, and the 100 sample data used for simulation. The true proportions are: 25.2% facies 1 (white); 51.1% facies 2 (grey); and 23.7% facies 3 (black). The sample data proportions are 25% facies 1, 51% facies 2, and 24% facies 3.

Figure 14 shows the scanned rock which was used as a training image for this example. The TI is 300 by 200 pixels and has 21.7% facies 1, 55.7% facies 2, and 22.6% facies 3. The global proportions and the overall look of the TI differ slightly from the true data, which is more realistic than the previous example. The TI is larger than the true image, which allows for good inference of long-range structure.

Using the Gibbs sampler algorithm, 100 realizations were simulated. The MPS template contained 14 four-point MPEs, five grid discretizations were used, the servosystem parameter was 0.8 and the noise reduction factor was 0.2. All 100 realizations were stopped by the first criterion of the number of changes leveling out. Calculation of the statistics took 6:00 (six minutes), which is significantly longer than the first example. This is because of the larger size of the problem, with a maximum matrix size (all classes in all MPEs informed) of $MK^N = 1134$ as opposed to the maximum possible size of 224 for only two facies. Simulating the 100 realizations took 2:35 on the same machine as before, a Dell Precision 670 with dual 3.2 GHz processors and 3 GB of RAM. Four of the realizations are shown in Figure 15.

While the realizations appear to have quite a bit of randomness, the features are similar to what is seen in the TI; the true data set is significantly cleaner and less random than the TI. The univariate proportions for the realizations are 23.4% facies 1, 54.9% facies 2, and 21.7% facies 3, which are close to the true and TI values. Because the true data and TI do not have the same proportions some variation is to be expected.

Figure 16 shows the probability of each of the three facies over the 100 realizations. Facies 1 appears to have “clumps” of high probabilities around the conditioning data; facies 3 shows, on average, the characteristic long, narrow bands. Facies 2 is what could be called the “background” facies. Figure 17 shows the indicator variograms for the three different facies in the X (east-west) and Y (north-south) directions for the sample data, TI, true data, and ten realizations. All of the variograms were reproduced reasonably well, with the realization variograms being closer to the true data than to the TI, which is most likely a data conditioning effect. All three facies demonstrate some level of cyclicity in the Y direction and this was reproduced to varying extents, although the TI has little of this feature compared to the true data.

To measure the effectiveness of the prediction of the proposed algorithm, an accuracy plot was calculated. This is shown in Figure 18. Each facies is plotted, with the predicted probability at each location on the X axis and the actual proportion of points at those locations with the given facies on the Y axis. A line with a 1:1 slope and lines 15% above and below this are also plotted. Only one point falls outside of the 15% error envelope, and most of the other points are close to the 1:1 line.

As a measure of the higher-order structure in the realizations, the MPS histograms were calculated (Boisvert, 2007). The MPS used were two cells by two cells square for a four-point statistic with $3^4 = 81$ classes. The histograms from the TI, true data, and realizations are shown in Figure 19. The TI and true data histograms are side-by-side bars; the mean realizations histogram values are shown as dots; and the 0.05 and 0.95 quantile class frequencies are shown as whisker plots. From Figure 19, it may be seen that the realization MPS histograms are very close to the true and TI histograms for the highest-frequency classes. The lowest frequency MPS classes have the greatest relative spread, but those classes which occur less than once out of every thousand points have less importance. The average sum of absolute differences between the realizations and the true data is 0.088, which is very good reproduction. As another higher-order statistical measure, the runs in the X and Y directions were calculated (Boisvert, 2007). The results of these calculations are shown in Figure 20. The runs are the probabilities of consecutive points of the same facies and are a good measure of the connectivity of a field. In Figure 20 it may be seen that the realizations match the true data runs distribution quite well in the X direction, and the TI runs distribution in the Y direction. In both cases all of the distributions are very close together, suggesting similar connectivity and therefore flowpaths.

Overall, the new Gibbs sampler algorithm seems to reproduce the complex features seen in the training image. The statistics are matched well.

Conclusions and Future Work

Compared to last year’s implementation (Lyster et al, 2006), the Gibbs sampler algorithm is greatly improved in both speed and reproduction of features. There are still some issues to be explored: further increases in computational efficiency could be implemented; the order in which the modifications to the estimated conditional probabilities are performed could be experimented with; the selection of the templates and the number and size of MPEs to use should be explored in greater detail; and exploration of the parameters such as the number of grids, servosystem factor, and noise reduction factor could be performed. As currently implemented the algorithm appears to be nearly ready for a full-scale case study using real data and proceeding from selection of appropriate training images through to response characterization such as grade above cutoffs or flow simulation.

References

- Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D. (1999) *LAPACK User's Guide, 3rd Ed.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 429 p.
- Boisvert, J.B. (2007) *Mineral Deposit Modeling with Pseudo-Genetically Constructed Training Images*. M.Sc. Thesis, University of Alberta, 72 p.
- Caers, J. (2001) Geostatistical Reservoir Modelling Using Statistical Pattern Recognition. *Journal of Petroleum Science and Engineering*, Vol. 29, No. 3, May 2001, pp 177-188.
- Caers, J. and Journel, A.G. (1998) Stochastic Reservoir Simulation Using Neural Networks Trained on Outcrop Data. *SPE Annual Technical Conference and Exhibition*, New Orleans, Oct. 1998, pp 321-336. SPE #49026.
- Casella, G. and George, E.I. (1992) Explaining the Gibbs Sampler. *The American Statistician*, Vol. 46, No. 3, Aug. 1992, pp 167-174.
- Deutsch, C.V. (1992) *Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (Well Test) Data*. Ph.D. Thesis, Stanford University, 304 p.
- Deutsch, C.V. (2002) *Geostatistical Reservoir Modeling*. Oxford University Press, New York, 376 p.
- Deutsch, C.V. and Journel, A.G. (1998) *GSLIB: Geostatistical Software Library and User's Guide, 2nd Ed.* Oxford University Press, New York, 369 p.
- Deutsch, C.V. and Tran, T.T. (2002) FLUVSIM: A Program for Object-based Stochastic Modeling of Fluvial Depositional Systems. *Computers & Geosciences*, Vol. 28, No. 4, May 2002, pp 525-535.
- Gelfand, A.E. and Smith, A.F.M. (1990) Sampling-Based Approached to Calculating Marginal Densities. *Journal of the American Statistical Association*, Vol. 85, No. 410, June 1990, pp 398-409.
- Geman, S. and Geman, D. (1984) Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No. 6, Nov. 1984, pp 721-741.
- Guardiano, F.B. and Srivastava, R.M. (1993) Multivariate Geostatistics: Beyond Bivariate Moments. Soares, A., Editor, *Geostatistics Troia '92*, Vol. 1, pp 133-144.
- Liu, Y. (2006) Using the Snesim Program for Multiple-Point Statistical Simulation. *Computers & Geosciences*, Vol. 32, No. 10, Dec. 2006, pp 1544-1563.
- Lyster, S., Leuangthong, O., and Deutsch, C.V. (2004a) Simulated Annealing Post Processing for Multiple Point Statistical Reproduction. *Centre for Computational Geostatistics*, No. 6, 15 p.
- Lyster, S., Ortiz, J.M. and Deutsch, C.V. (2004b) MPASIM: A Multiple Point Annealing Simulation Program. *Centre for Computational Geostatistics*, No. 6, 18 p.
- Lyster, S., Deutsch, C.V., and Dose, T. (2006) A New MPS Simulation Algorithm Based on Gibbs Sampling. *Centre for Computational Geostatistics*, No. 8, 26 p.
- Lyster, S., and Deutsch, C.V. (2006) Short Note: An Entropy-Based Approach to Establish MPS Templates. *Centre for Computational Geostatistics*, No. 8, 10 p.
- Lyster, S., and Deutsch, C.V. (2007) Artifacts Near Conditioning Data in MPS Gibbs Sampling. *Centre for Computational Geostatistics*, No. 9, awaiting publication.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953) Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, Vol. 21, No. 6, pp 1087-1091.
- Ortiz, J.M. (2003) *Characterization of High Order Correlation for Enhanced Indicator Simulation*. Ph.D. Thesis, University of Alberta, 255 p.
- Srivastava, M. (1992) Iterative Methods for Spatial Simulation. *Stanford Center for Reservoir Forecasting*, No. 5, 24 p.
- Strebelle, S. and Journel, A.G. (2000) Sequential Simulation Drawing Structures From Training Images. Kleingeld, W.J. and Krige, D.G., Editors, *6th International Geostatistics Congress*, 12 p.
- Strebelle, S. and Journel, A.G. (2001) Reservoir Modeling Using Multiple-Point Statistics. *SPE Annual Technical Conference and Exhibition*, New Orleans, Oct. 2001, 11 p. SPE #71324.
- Strebelle, S. (2002) Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics. *Mathematical Geology*, Vol. 34, No. 1, Jan. 2002, pp 1-21.

	Estimates			Error Variances		
	Facies 1	Facies 2	Facies 3	Facies 1	Facies 2	Facies 3
Global values	0.5003	0.3963	0.1034	0.2500	0.2393	0.0927
2006	0.3414	0.9589	-0.3003	0.1729	0.1299	0.0872
2007	0.0302	0.9728	-0.0031	0.0444	0.0477	0.0205

Table 1: Comparison of using only those MPEs with indicator values of 1 (2006) and all MPEs (2007).

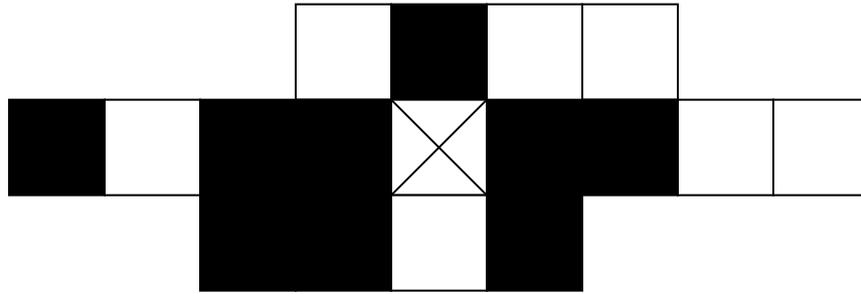


Figure 1: This arrangement of points may not appear in the TI.

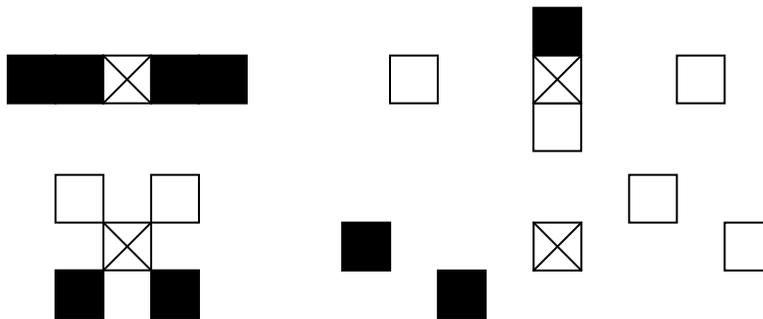


Figure 2: The template and data from Figure 1, divided into four discrete MPEs.

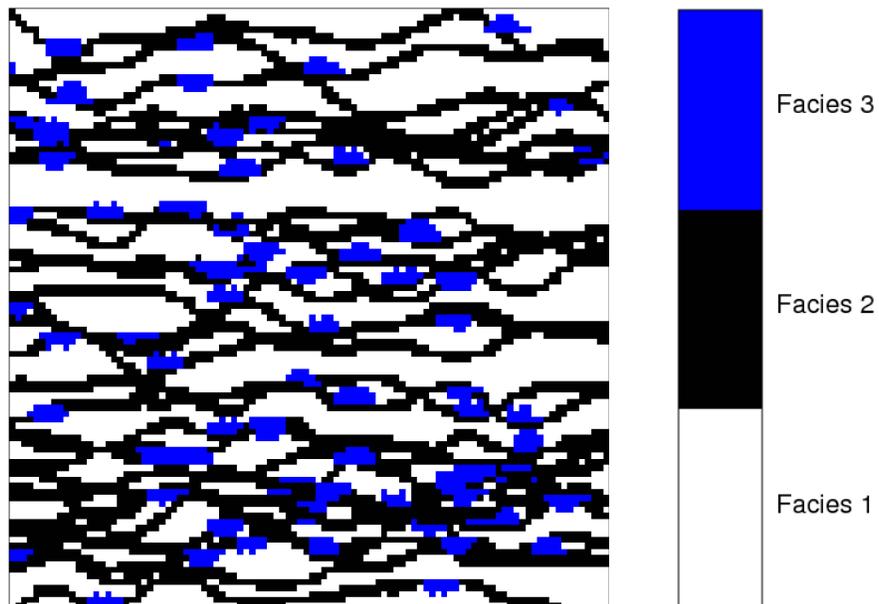


Figure 3: A training image used in an example for estimation of a ccdf.

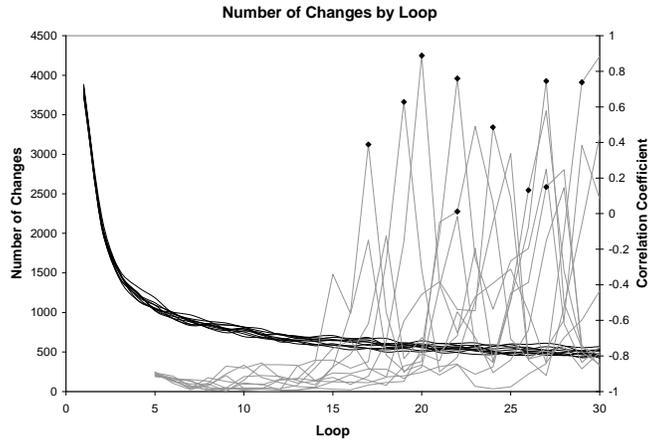


Figure 4: Number of changes (black line) in ten realizations of a simulated model for each loop over all locations, and the correlation coefficient for the last five loops (grey line). The black dots identify the first positive correlation for each realization.

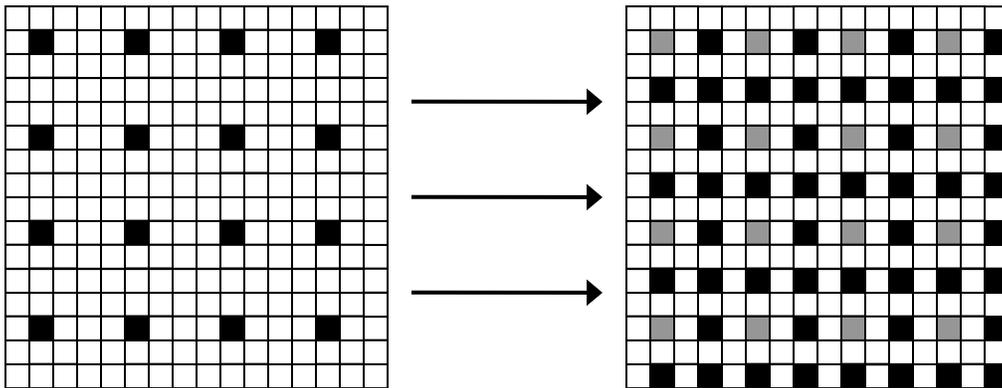


Figure 5: A small example of multiple grids. Left: A 4x4 grid discretization. Right: a 2x2 grid discretization. Locations to be simulated are black, those frozen in place from a previous grid are grey.

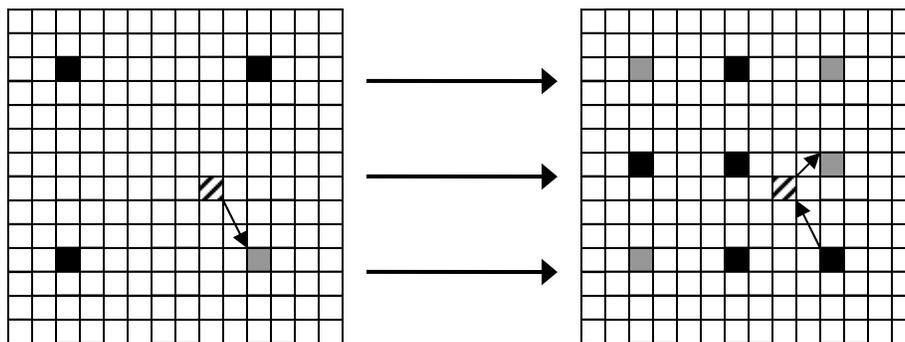


Figure 6: An example of how conditioning data are assigned to the grid at different discretizations.

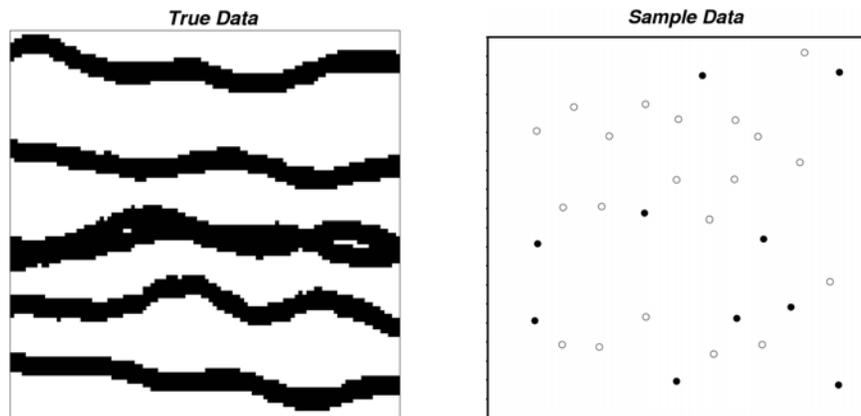


Figure 7: The 100x100 true field and 30 samples for the fluvial channel example.



Figure 8: The 200x100 TI used for the fluvial channel example.

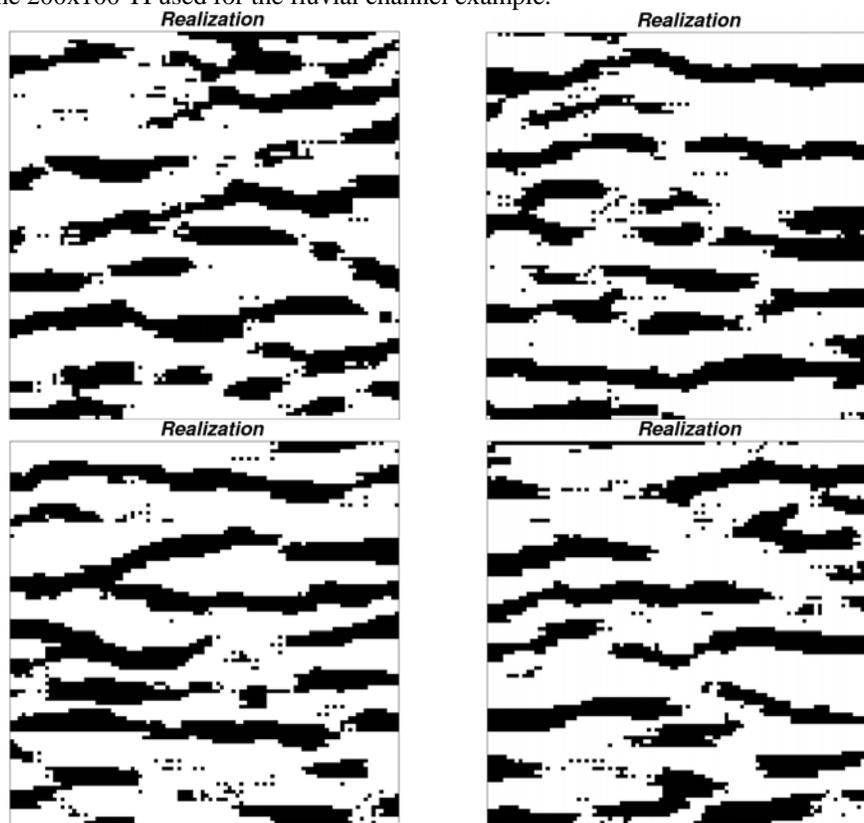


Figure 9: Four 100x100 realizations created with the proposed Gibbs sampler algorithm.

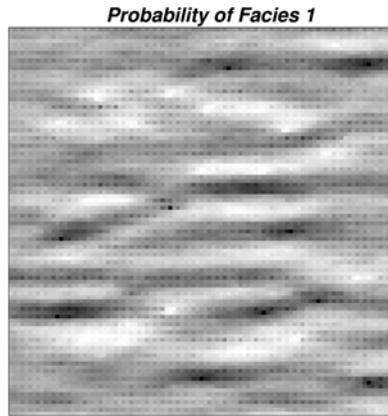


Figure 10: The probability of facies 1 (channel) over 100 realizations.

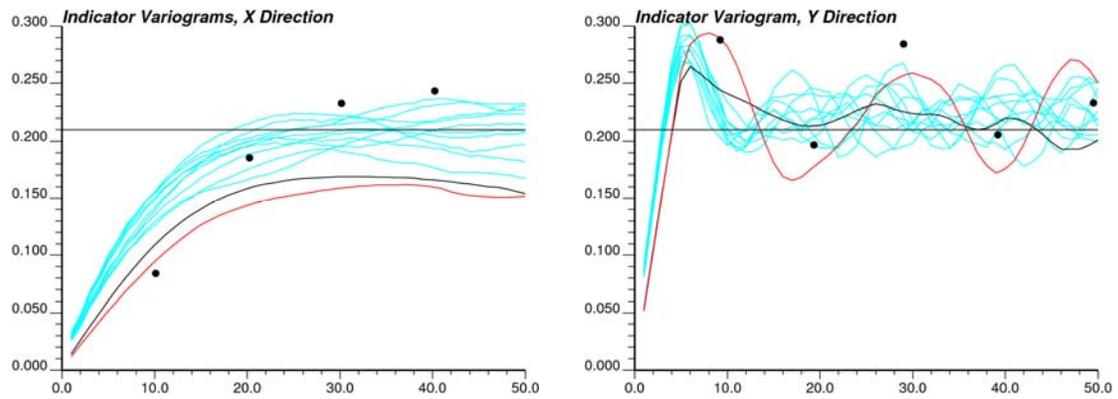


Figure 11: The indicator variograms for the sample data (black points), training image (black line), ten realizations (blue lines), and true data (red line).

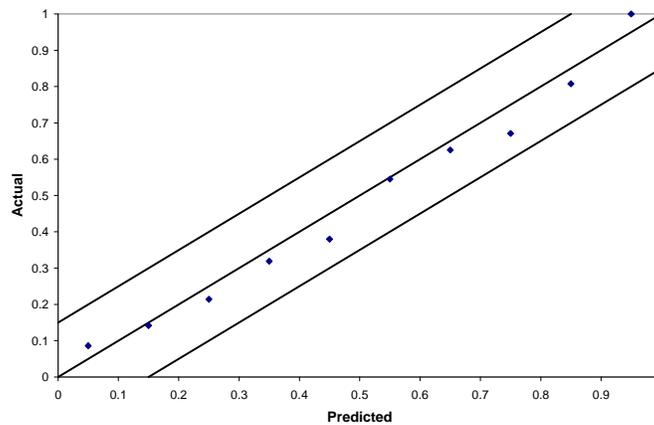


Figure 12: An accuracy plot for 100 realizations compared to the true data, with +/- 15% intervals shown.

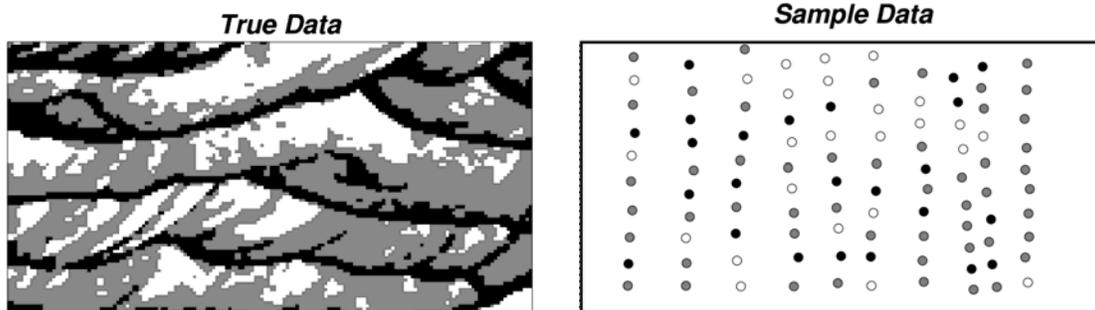


Figure 13: The 164x85 true field and 100 samples for the eolian sandstone example. (Deutsch, 1992)

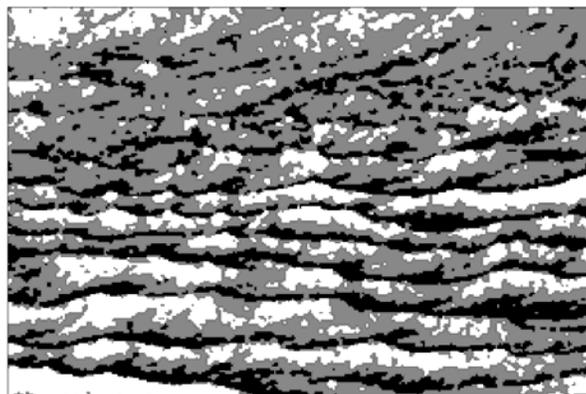


Figure 14: The 300x200 training image used for the eolian sandstone example. (Deutsch, 1992)

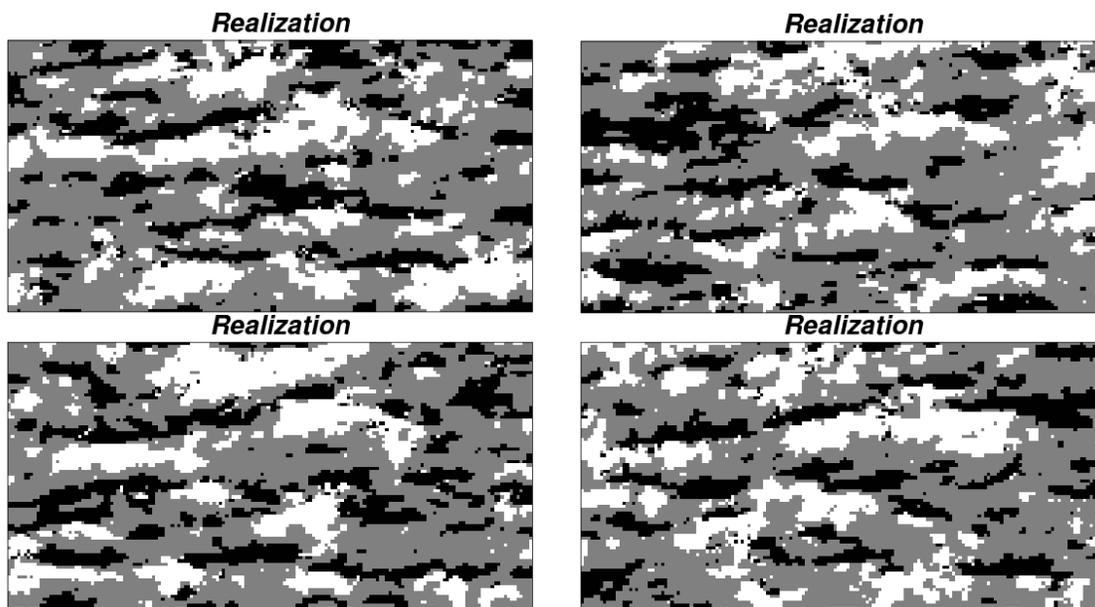


Figure 15: Four 164x85 realizations created with the proposed Gibbs sampler algorithm.

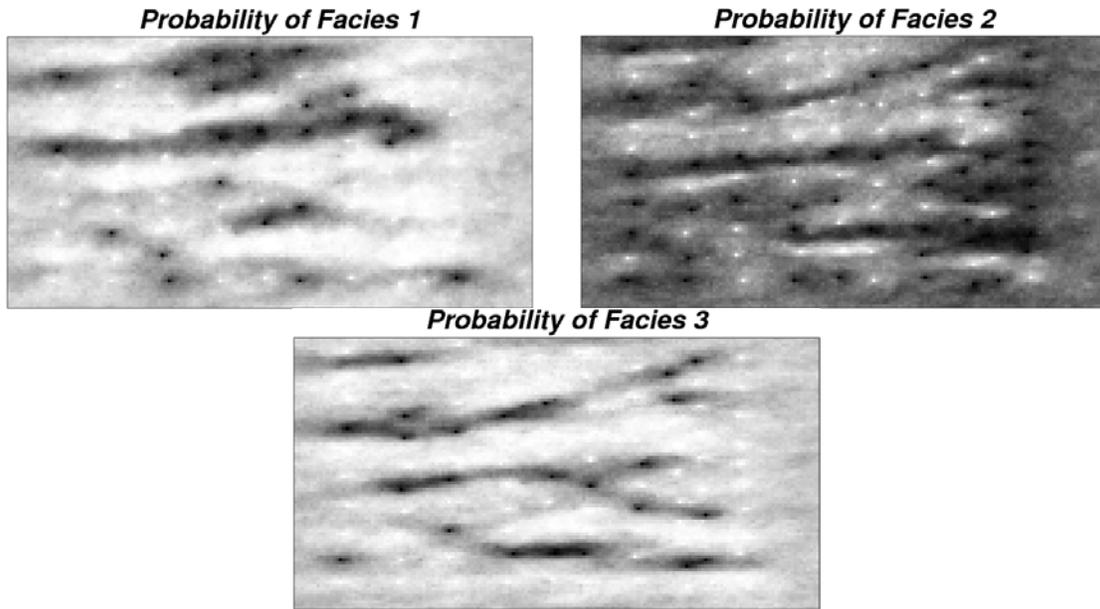


Figure 16: The average probabilities of each facies over 100 realizations. Facies 1 is white, facies 2 is grey, and facies 3 is black.

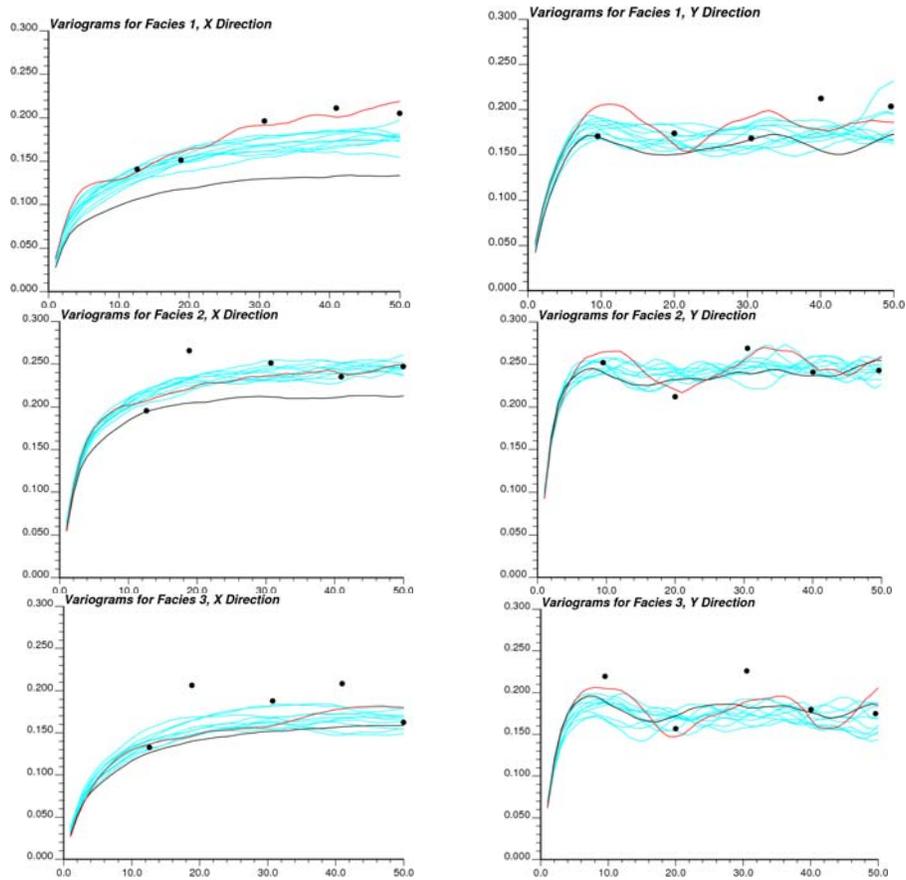


Figure 17: The indicator variograms for the sample data (black points), training image (black line), ten realizations (blue lines), and true data (red line). Facies 1 is white, facies 2 is grey, and facies 3 is black.

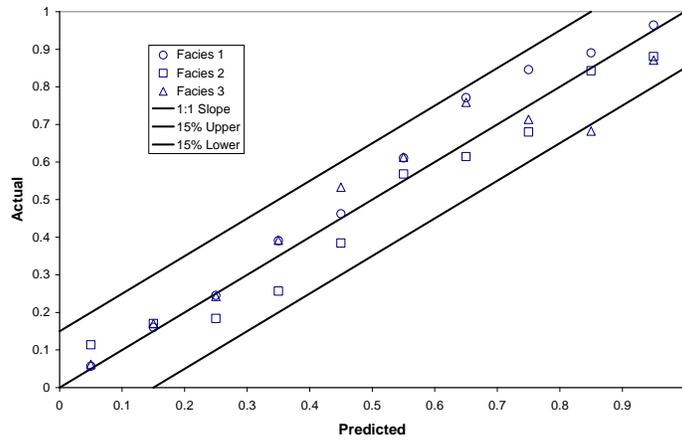


Figure 18: An accuracy plot for 100 realizations compared to the true data, with +/- 15% intervals shown. Facies 1 is white, facies 2 is grey, and facies 3 is black.

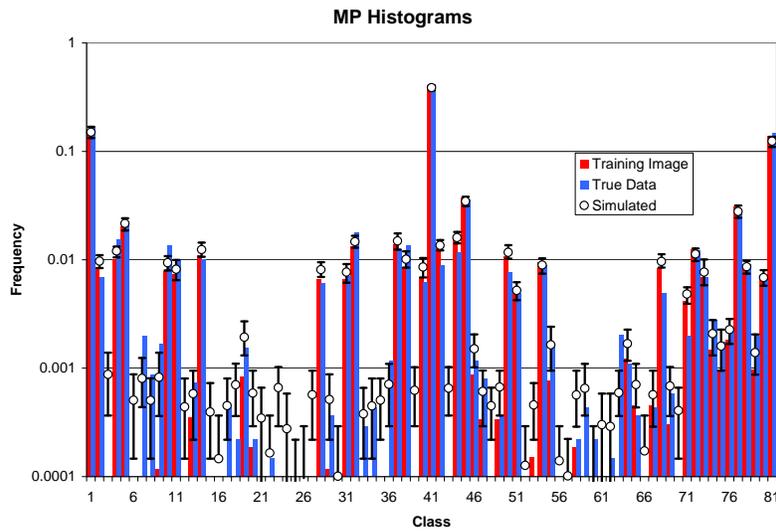


Figure 19: Four-point, 2x2 MPS histograms for the TI (red), true data (blue) and 100 realizations (points, with 90% intervals shown).

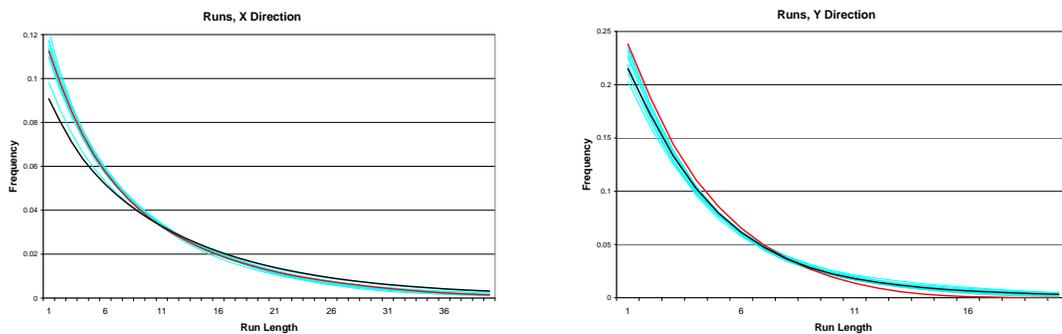


Figure 20: Runs distributions for the TI (black line), true data (red line) and ten realizations (blue lines) for the eolian sandstone example.