

MPASIM: A Multiple-Point Annealing Simulation Program

Steven Lyster¹, Julián M. Ortiz², and Clayton V. Deutsch¹

¹Department of Civil & Environmental Engineering
University of Alberta

²Departamento de Ingeniería de Minas
Universidad de Chile

Abstract

Multiple-point statistics have the potential to improve simulation of complex geologic structures featuring high-order relations. Reproduction of multiple-point statistics can be difficult; simulated annealing is one potential method for honouring multiple-point statistics in practical applications. This short note discusses a program that was written to explore the potential of multiple-point simulated annealing with modern computers and recent insight into multiple-point statistics. This program can be used for any size of training image, simulated field, and anisotropic prismatic multiple-point statistic in three dimensions. A new approach to the simulated annealing schedule was used to increase the speed of simulation. Several examples are presented.

Introduction

Reproduction of multiple-point (MP) statistics for geostatistical modeling can be problematic. A statistic commonly used to characterize the behaviour of multiple points is the MP histogram. If each possible combination of categorical data within a multiple point configuration is coded with an index, then the MP histogram is defined by plotting the frequency of occurrence of these indexes. For a continuous variable, a similar coding can be done by defining indicators for different thresholds.

In the case of the MP histogram many classes may be represented for a relatively small size of MP configuration. The exact replication of the histogram in a given realization is not desirable; rather, the statistics should be reproduced on average over many realizations. Whether to allow classes that occur in the training image (TI) to not occur in a given realization or to allow new classes to appear is a difficult decision. Establishing the features and statistics that are essential to the essence of the TI is more important than duplicating the TI exactly [5].

Simulated annealing (SA) can be a powerful tool for many applications [1,3]. Iterating over many perturbations can result in an optimized solution if an appropriate objective function is chosen and the annealing schedule is properly set. The ability of SA to accept unfavourable perturbations allows the simulation to escape from local minima and achieve a better solution. As SA is an iterative approach to simulation, it can also be used to post-process images created by other means. This combination of different approaches can lead to better results than one method could produce by itself.

MPASIM Program

To fully explore how effectively MP statistics can be used with SA, the program MPASIM (Multiple-Point Annealing SIMulation) was written. This program uses only a single MP pattern, specified by the user, for simulation. The use of a single statistic eliminates any interference or inconsistencies in the process from other components to the objective function such as a two-point variogram or covariance. This design choice results in realizations being returned by the program that may be analyzed consistently to measure the goodness and relevance of the MP objective function. It must be mentioned, however, that if all statistics (histogram, variograms, and MP histogram) are calculated from the same exhaustive training image, no inconsistencies should occur.

MPASIM can simulate any size of field the user desires, with the statistics calculated from any size of TI. Alternatively, the MP histogram may be read in from a file (used if the MP histogram is to be scaled to a different single-point distribution). The number of facies K and points N of the pattern for the MP simulation are specified by the user and can be any number up to the limits of array storage. The K^N classes in the histogram cannot be greater than $2^{31}-1$.

The particular N -point statistic used is a 3-dimensional rectangular prism shape, with size in the X, Y, and Z directions specified by the user; isotropy is not required. For 2-dimensional simulations a size of 1 in the Z direction is used. The statistic is only limited in size by the overall limit of K^N .

All input and output files are in standard GSLIB format. The program MPASIM requires only the file MPASIM.FOR to run. All subroutines and functions are included. MPASIM prompts for a parameter filename when run; if none is entered and MPASIM.PAR does not exist a default parameter file will be created. The parameters for the program are given below:

Line	START OF PARAMETERS:	
1	1	\inopt Input option (0=histogram, 1=TI)
2	input.dat	\infl File with data
3	1	\trcol Column for training image data (inopt=1)
4	200 100 1	\trxsiz,ysiz,zsiz Size of training image (inopt=1)
5	1 2	\coll col2 Columns for histogram class and frequency (inopt=0)
6	2 2 1	\dx dy dz Statistic dimensions in x,y,z directions
7	5	\nc Number of rock codes
8	1 2 3 4 5	\incode(i) Rock codes
9	200 100 1	\xsiz ysiz zsiz Size of field to simulate
10	1	\nreal Number of realizations
11	0	\cond Use conditioning data? (1=yes,0=no)
12	condfile.dat	\condfl File for conditioning data (cond=1)
13	1 2 0 3	\condx,y,z,rc Columns for conditioning data (cond=1)
14	0	\pp Post process? (1=yes, 0=no)
15	ppinput.dat	\ppfl File with data for post processing (pp=1)
16	0	\nfreeze Number of rock codes to freeze (pp=1)
17	1 2	\freeze(i) Codes to freeze (pp=1)
18	69069	\rseed Random number seed
19	mpasim.dbg	\dbgfl File for debugging output
20	mpasim.out	\outfl File for output

The first five parameter lines define the input options from which the statistics will be calculated. The *inopt* gives the user the option to read the multiple-point histogram from a file or calculate them from a TI file. *infl* is the file containing the data, whether histogram or TI. *trcol* is the column in *infl* containing the rock codes. The size of the TI is defined by *trxsiz*, *trysiz*, and *trzsiz*.

If the histogram is to be read directly from a file rather than calculated, *col1* is the column in *infl* containing the class numbers and *col2* is the column containing the frequencies.

After the input, lines 6 through 8 define the statistic to be used. The *dx*, *dy*, and *dz* parameters are the size of the statistic in the principal directions. The configuration of locations in the MP histogram used in MPASIM is always a rectangular prism shape, and anisotropy is allowed. The number of rock codes is defined by *nc* and the codes themselves are input as *incode(i)*, where $i=1, \dots, nc$. This allows the rock codes to be any integer values, not just sequential values starting at 1.

Lines 9 and 10 define the simulation. The field for simulation is defined by *xsiz*, *ysiz*, and *zsiz*. The number of realizations to perform is read in as *nreal*.

The three lines after that (11 through 13) define the option to use conditioning data. *cond* is the option flag of whether or not to use conditioning data. The file containing the data is input as *condfl*, and the columns for the x, y, and z coordinates and rock codes are identified by *condx*, *condy*, *condz*, and *condrc*.

Parameter lines 14 and 15 deal with post-processing in MPASIM. This option is turned on by the flag *pp* and the file containing the post-processing data is *ppfl*. In addition to post-processing, rock codes may be frozen if desired. This allows a stepwise process to be used to simulate one rock code at a time, that is, the simulation proceeds in a binary mode, modelling one rock code versus all others, then the rock code is frozen and a second rock code is simulated versus the remaining ones, and so on. The *nfreeze* parameter is the number of rock codes to freeze; it can be zero if freezing is not desired. *freeze(i)* for $i=1, \dots, nfreeze$ stores the rock codes to be frozen. The frozen codes will not be selected for a perturbation, and will not be perturbed. They will be treated like conditioning data.

The last three lines of the parameter file read in the random number seed to be used for simulation, *rseed*, and the debugging and output files, *dbgfl* and *outfl*. The debugging file outputs the initial objective as well as the number of total perturbations and perturbations accepted at each temperature decrease.

Simulation Steps

An outline of the steps taken for simulation follows. These steps all occur after the parameter file has been successfully read in.

Get the Target Histogram from File or Training Image

Depending on the input option, MPASIM will either get the desired MP histogram from a file or calculate it from a given training image file. An input histogram file needs to have the class number in one column and the frequency of this class in another. This option is used if the MP distribution is to be scaled to match a different single-point histogram.

MPASIM can also calculate the target MP histogram from a TI file. The file must be in standard GSLIB format and be of the size specified in the parameters. There is no difference in the option used to obtain the statistics once simulation proceeds.

Read Conditioning Data and Develop Random Path

If conditioning data are to be used, it is read in from the data file at this time. The X, Y, Z coordinates and rock code in separate columns are required. The conditioning data are counted and the nodes they occupy are stored so they will not be perturbed.

Once all conditioning data have been read, the distances from all remaining nodes to be simulated to the nearest conditioning data are found. Each of these has a small random component added to it to prevent identical values. Then, these distances are sorted and the points given an order in the path starting closest to the conditioning data and moving away. This is stored as the path to visit the nodes, and the order of all perturbations will follow it.

Weight the Points

To prevent edge effects and better honour the conditioning data, all points in the simulation field are given a weight. Every node is assigned a weight of 1 initially. Then, if a node is close to conditioning data (within the extents of the MP configuration) the weight of the node is doubled.

The edge effect correction is somewhat more complicated. The weights of all points are multiplied by a factor such that the perturbation of any node will have the same potential to change the objective function. For example, a point at the edge of the simulation field will have its weight multiplied by the size of the MP configuration in the direction normal to the edge the node is at. If the node is slightly in from the edge, the factor will be proportionally less. If a node is at or near two or more edges, its weight will be multiplied by a factor for each edge. Therefore, if a 25-point statistic is used all points will have the potential to change the objective function by the equivalent of an increase or decrease of 25 occurrences in the various classes whose frequency would be affected. This weighting is illustrated in two two-dimensional examples in Figure 1.

4	2	2	2	2	2	2	4
2	1	1	1	1	1	1	2
2	1	1	1	1	1	1	2
2	1	1	1	1	1	1	2
2	1	1	1	1	1	1	2
2	1	1	1	1	1	1	2
2	1	1	1	1	1	1	2
4	2	2	2	2	2	2	4

9	4.5	3	3	3	3	4.5	9
4.5	2.25	1.5	1.5	1.5	1.5	2.25	4.5
3	1.5	1	1	1	1	1.5	3
3	1.5	1	1	1	1	1.5	3
3	1.5	1	1	1	1	1.5	3
3	1.5	1	1	1	1	1.5	3
4.5	2.25	1.5	1.5	1.5	1.5	2.25	4.5
9	4.5	3	3	3	3	4.5	9

Figure 1: Left: An 8x8 field showing the MPASIM weights for each point using a 2x2 statistic. Right: The weights for the same field using a 3x3 statistic.

The weights are applied to the change in the objective function for the purpose of determining acceptance of the perturbation. The local updating of the objective function and MP histogram class frequencies are not affected.

As an alternative to weighting the grid could be “wrapped”, meaning that each point at an edge would be assumed to be next to the point on the opposite edge. Using this approach would be very convenient for eliminating edge effects but would not reflect reality. Also, this process could be computationally expensive, as a check for grid wrapping would be required at every perturbation. This option was not implemented in MPASIM.

Develop Initial Image

Once the weighting has been determined, the simulation may proceed. The initial image may be generated randomly by drawing from the single-point distribution at every node.

Alternatively, a previously generated image may be used as a starting point for the simulation. This can improve the speed of the simulation by reducing the number of perturbations necessary. Post-processing will be discussed in more detail later. If post-processing of an existing image is to be used, a file containing the image in GSLIB format must be provided. The filename is specified in the parameter file.

Calculate Initial Objective

In MPASIM the objective function is defined as:

$$O = \sum_{j=1}^{K^N} [f_j - f_j^*]^2 \quad (1)$$

where f_j is the target frequency of class j of the MP histogram, calculated from the TI or read from the histogram file, and f_j^* is the realization frequency of class j . The realization frequency is calculated by cycling through all points in the simulation field that can exist as the lower-left corner of an occurrence of the N -point statistic and counting the occurrences of each class j . The number of times each class occurs is then divided by the total number of N -point statistics in the field to give f_j^* .

Determine Initial Temperature (First Realization Only)

The initial temperature for SA is set automatically by MPASIM, simplifying the simulation process for the user. To find the initial temperature, all nodes to be simulated are cycled through along the path determined earlier and perturbed once. The perturbation is not checked for acceptance, however; the node, objective function, and MP histogram remain unchanged. The maximum absolute value change in the objective function is stored and used to determine the initial temperature.

Selection of the initial temperature differs based on the initial image option. If the initial image is randomly generated, the initial temperature is set as ten times the maximum absolute value change in the objective function. If post-processing, the temperature is set lower both to preserve the initial image characteristics better and to improve speed of simulation. In this case, the initial temperature is one-tenth the maximum absolute value objective function change.

The initial temperature is retained for all realizations. While it would be slightly different if calculated for subsequent realizations, the difference is not considered to be important.

Perturb Nodes

The random path developed earlier is followed and nodes perturbed. The MP histogram is updated for each accepted perturbation by a factor δ , which is the value of changing a class j in the histogram by one occurrence. The objective function is updated locally using the equation

$$\begin{aligned} \Delta O = & \sum_{n=1}^N \left[f_{j_{new}}(n) - (f_{j_{new}}^*(n) + \delta) \right]^2 - \left[f_{j_{new}}(n) - f_{j_{new}}^*(n) \right]^2 \\ & + \left[f_{j_{old}}(n) - (f_{j_{old}}^*(n) - \delta) \right]^2 - \left[f_{j_{old}}(n) - f_{j_{old}}^*(n) \right]^2 \end{aligned} \quad (2)$$

where j_{new} is the class that would have one occurrence added because of an accepted perturbation and j_{old} is the class that would have one occurrence taken away. Note that equation (2) must be summed over N points because a single perturbation affects N arrangements of the MP statistic being used (unless an edge point is being perturbed, which is why those points are given extra weighting).

While nodes are being perturbed the accepted perturbations are counted. The total number of perturbations is not counted; rather, the random path is rotated through ten times before the conditions for lowering the temperature are checked. Attempting ten perturbations at every node, following the path, is called a *cycle* in the MPASIM program. Performing complete cycles without checking for convergence improves speed without negatively affecting the quality of the simulation.

Check Conditions for Lowering Temperature

Because of the new flavour of SA used by MPASIM, the conditions for lowering the temperature are slightly different than the typical annealing schedule used. After a cycle has been completed, the following relation is checked:

$$\frac{N_{accept}}{N_{perturb}} > \frac{0.9}{N_{cycles}} \quad (3)$$

where N_{accept} is the number of perturbations accepted, $N_{perturb}$ is the total number of perturbations attempted, and N_{cycles} is the number of cycles completed. This relation, if set as an equation, corresponds to an acceptance rate of 90% for one completed cycle then lowering the acceptance rate with each cycle completed as shown in Table 1. If 10 cycles are completed, the temperature is lowered regardless of the number of perturbations accepted; the number of times this occurs is counted.

<u>Number of Cycles Completed</u>	<u>Acceptance Rate for Lowering Temperature</u>
1	90%
2	45%

3	30%
4	22.5%
5	18%
6	15%
7	12.86%
8	11.25%
9	10%
10	Lowered regardless

Table 1: Number of cycles completed with the corresponding acceptance rate required for lowering the temperature.

If the left side of relation (3) is greater than the right after a cycle has been completed, then the temperature will be lowered. For simplicity this relation can be simplified using the equations

$$N_{perturb/cycle} = 10 \times (xsiz \times ysiz \times zsiz - N_{cond}) \quad (4)$$

$$N_{perturb} = N_{cycles} \times N_{perturb/cycle} \quad (5)$$

where $N_{perturb/cycle}$ is the number of perturbations attempted in once cycle, N_{cond} is the number of conditioning data used in the simulation, and $xsiz$, $ysiz$, $zsiz$ are the size of the simulation field, in nodes, in each principal direction. Combining relation (3) with equations (4) and (5) gives the new relation

$$N_{accept} > 9 \times (xsiz \times ysiz \times zsiz - N_{cond}) \quad (6)$$

which is the relation checked by MPASIM after each cycle and yields the critical acceptance rates given in Table 1; if the acceptance rate is greater than the critical rate the temperature is lowered. The use of this unconventional approach to SA allows MPASIM to skip many checks and counters and increases speed of simulation. This method gives a similar number of perturbations attempted and accepted as standard SA with a K_{max} of 100 times the number of grid nodes and K_{accept} of 10 times the number of grid nodes.

When the criterion for lowering the SA temperature has been met after a cycle, the temperature is multiplied by a λ value of 0.2. Typically the λ value is 0.1, but in MPASIM the decision was made to slow down the temperature decrease to prevent severe edge effects and allow for a more optimal realization. This decision was made to allow for more flexibility; rather than adjust the annealing schedule for a different TI or MP histogram, the simulation is allowed to cool slowly. This should lead to acceptable convergence in most cases, without requiring an experienced user to set the parameters.

Finish the Realization

Due to the huge number of classes usually found in a MP histogram, convergence to some low objective function is very unlikely. This could be linked to a problem with the assumption of ergodicity: the algorithm may work properly, but it would be needed a very large domain to ensure that all combinations of the multiple point pattern are considered during the simulation process. Rather than waste CPU time on checking for convergence, a realization is stopped only when ten cycles have been completed before lowering the temperature three times. This is the

same as standard SA using a stopping number of three. After simulation is ended, the realization is written to the output file and a new one is started if necessary.

The resulting realizations should be checked for histogram and variogram reproduction, as a standard procedure, to ensure that minimum standards are met.

Post-Processing and Stepwise Simulation

Post-processing of an existing image can have many benefits in SA. The input image could honour other statistics not directly included in the SA objective function such as the variogram or covariance. Because these lower-order statistics are contained implicitly within the MP histogram, they will be better honoured by the SA process. Starting with an image closer to the TI reduces the number of perturbations required to achieve a suitable output realization and improves matching of the MP statistics.

Another benefit of post-processing is the ability to perform a stepwise simulation. This involves simulating an image with several rock codes combined, such as the background rock code vs. all remaining codes. Then one more facies type may be added at a time to the realization, freezing most of the previously simulated facies (i.e., the background facies). An example of this process is shown in Figure 3. The training image, as well as progression from first simulation to last, is shown. All of the stepwise simulations were done using a 3x3 square pattern and there are 5 different facies represented.

Examples

Two Dimensions, Three Rock Codes, Nine Points

Shown in the top left of Figure 4 is a two-dimensional cross-sectional realization from FLUVSIM featuring many fluvial channels. In the top right is one realization produced by MPASIM using a 3x3 MP pattern. Ten realizations took 7 minutes on a Pentium 4 2.0 GHz laptop with 640 MB of RAM. The TI and all realizations are 111 pixels in the X (horizontal) direction and 50 pixels in the Y (vertical) direction.

Below these images are six indicator semivariograms: two for each facies type, one in the X direction and one in the Y direction. Facies type 0 is the background brown colour, facies 1 is the yellow channels and facies 2 is the red. The black lines with points in the semivariograms plots are derived from the TI; the ten red lines each come from a different realization. The semivariograms show that even though the SA objective function contained only MP statistics, the two-point relations were captured quite well.

Two Dimensions, Two Rock Codes, Various Points

Figure 5 shows a comparison between simulations using the same TI but different MP statistics. The TI is shown in Figure 5a; it is 64x64 pixels in size and features 4x4 alternating black and white squares. Figures 5b, 5c, and 5d show MPASIM realizations using 2x2, 3x3, and 4x4 MP patterns respectively. The time required to perform ten realizations for each of these statistics is shown in Table 2. The simulations were done on a Pentium 4 1.8 GHz computer with 512 MB of RAM.

Figure	Pattern	Number of points, N	Time for 10 Realizations
5b	2x2	4	1:34
5c	3x3	9	5:09
5d	4x4	16	13:16

Table 2: Time to perform the simulations shown in Figure 5.

The higher-order statistics produce better results. The time required also increases significantly as N is raised. The balance between reproduction of structures and speed of simulation would be a consideration in practice.

To better understand the relationship between N , size of the grid being simulated, and time required, numerous simulations were performed using the TI shown in Figure 5a and various MP histogram pattern sizes. The simulations were carried out using 2x2, 3x3, and 4x4 arrangements. Figure 2 shows the results. The relationship between the size of the simulation grid and time required is remarkably linear. As expected, as N increases the time increases at a nearly exponential rate.

This is a simple example with two rock codes. If more rock codes are used, the time requirements may increase with K^N instead of N . Measuring the rate of increase is difficult as adding more facies also increases the complexity of the image; as more complex images take longer to simulate the measurements would be biased.

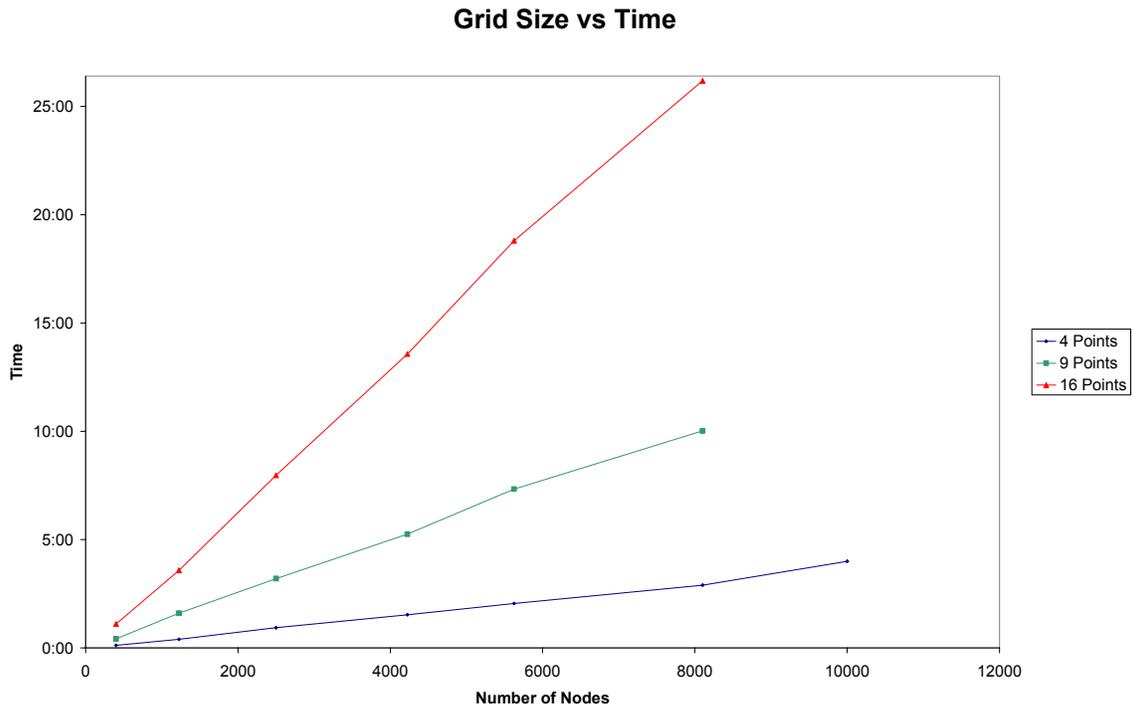


Figure 2: Number of grid nodes simulated with MPASIM versus

Three Dimensions, Three Rock Codes, Eight Points

Figure 6 shows sections of a TI generated by FLUVSIM that is 100x100x25 nodes. There are three rock codes, 0, 1, and 2, represented by brown, yellow, and red respectively. This TI was used as input for MPASIM to perform a simulation. The simulation used an 8-point MP histogram in a 2x2x2 configuration. Sections of the MPASIM output are shown in Figure 7. The simulation took 49 minutes to complete; if more realizations were performed in one run of the program the average time would be less than this due to the time taken to develop a random path, which is only done once.

The long-range structures and continuity of the channels are not captured well by the small pattern size used. The short-range arrangement of channels does appear, though there is some noise. The relation between the yellow and red facies was captured quite well. As MP statistics implicitly contain lower-order relations as well, the single-point histogram should be matched closely even though it is not directly included in the objective function. In the TI from FLUVSIM, the distribution of facies was: Facies 0, 60.1%; Facies 1, 22.6%; Facies 2, 17.3%. The MPASIM realization had a distribution of: Facies 0, 63.0%; Facies 1, 21.2%; Facies 2, 15.6%. The histogram is not exact but is quite close. Over many realizations, the single-point distribution should be matched.

To test the quality of simulation, indicator variograms were found for each facies in the X, Y, and Z directions. These variograms are shown in Figure 8. The red solid lines represent the indicator variograms for the TI, the black dashed lines the simulation. All of the indicator variograms are normalized, with the theoretical sill of 1.0 plotted.

The variograms show that the continuity in the Y direction is reproduced quite well. From the sections, it can be seen that the Y direction has the most definite relationships between the facies, while the X and Z directions feature much randomness and complex correlation. This is a possible cause for the good reproduction of the variogram.

As another test of the quality of simulation, the program FLOWSIM was used to determine the effective permeabilities of the TI and simulated realization in the X, Y, and Z directions, both for the overall permeability of the fields and for 5x5 blocks within the fields. Facies 0 was given an arbitrary permeability of 1, facies 1 a permeability of 100, and facies 2 a permeability 1000.

For the FLUVSIM TI the effective permeabilities were $K_{eff_x} = 25.8$, $K_{eff_y} = 122.3$, and $K_{eff_z} = 6.6$. The values for the MPASIM realization were $K_{eff_x} = 25.6$, $K_{eff_y} = 67.1$, and $K_{eff_z} = 8.9$. The X direction permeability is likely reproduced so well because the simulation structures are the shortest range, meaning a small MP histogram can capture the connectivity between the facies better. The Y direction does not have the smooth flow and continuous connections between the structures, despite the fact the variograms are reproduced well. In the vertical (Z) direction, the simulation appears to have resulted in slightly more continuity than is present in the TI, as is evident from the lower variogram values and higher K_{eff_z} .

The 5x5 block permeabilities are shown in histograms in Figure 9. The X direction once again has the best reproduction of flow, with mean K_{eff_x} values for the TI and simulation of 61.1 and 61.2 respectively. The K_{eff_y} and K_{eff_z} mean values are too low for the simulation, 127.1 compared to 137.1 in the Y direction and 25.5 compared to 35.5 in the Z direction. There is, however, good representation of the shape of the distribution except for there being too many very low values. This would suggest there is not enough connectivity in the Y and Z directions; for the Z direction this is contradictory to the overall K_{eff_z} . A possible explanation for this is that the higher

permeability 5x5 blocks are too continuous relative to one another, causing vertical flow channels that run the entire extent of the field while overall there are too few high permeability 5x5 blocks.

Conclusions

Simulated annealing using multiple point statistics is a potentially powerful tool. This method can reproduce complex structures that traditional two-point statistics do not adequately model. The ease of use of this technique when compared to complex, input-dependent techniques such as object-based modeling makes it an attractive compromise between simplicity and effectiveness. Currently, the realizations should be checked by other methods such as variogram analysis or connectivity function before being used further. Additional work streamlining the algorithm and improving reproduction of the MP histogram is required before this approach becomes practical for most users.

References

1. C.V. Deutsch. *Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (Well Test) Data*. PhD thesis, Stanford University, Stanford, CA, 1992.
2. C.V. Deutsch and A.G. Journel. *GSLIB: Geostatistical Software Library and User's Guide*. Oxford University Press, New York, 2nd edition, 1998.
3. C.V. Deutsch. *Geostatistical Reservoir Modeling*. Oxford University Press, New York, 2002.
4. J.M. Ortiz. Characterization of High Order Correlation for Enhanced Indicator Simulation. PhD thesis, University of Alberta, Edmonton, AB, 2003.
5. J.M. Ortiz. Selected aspects of multiple-point statistics. In *Centre for Computational Geostatistics*, Volume 5, Edmonton, AB, 2003.

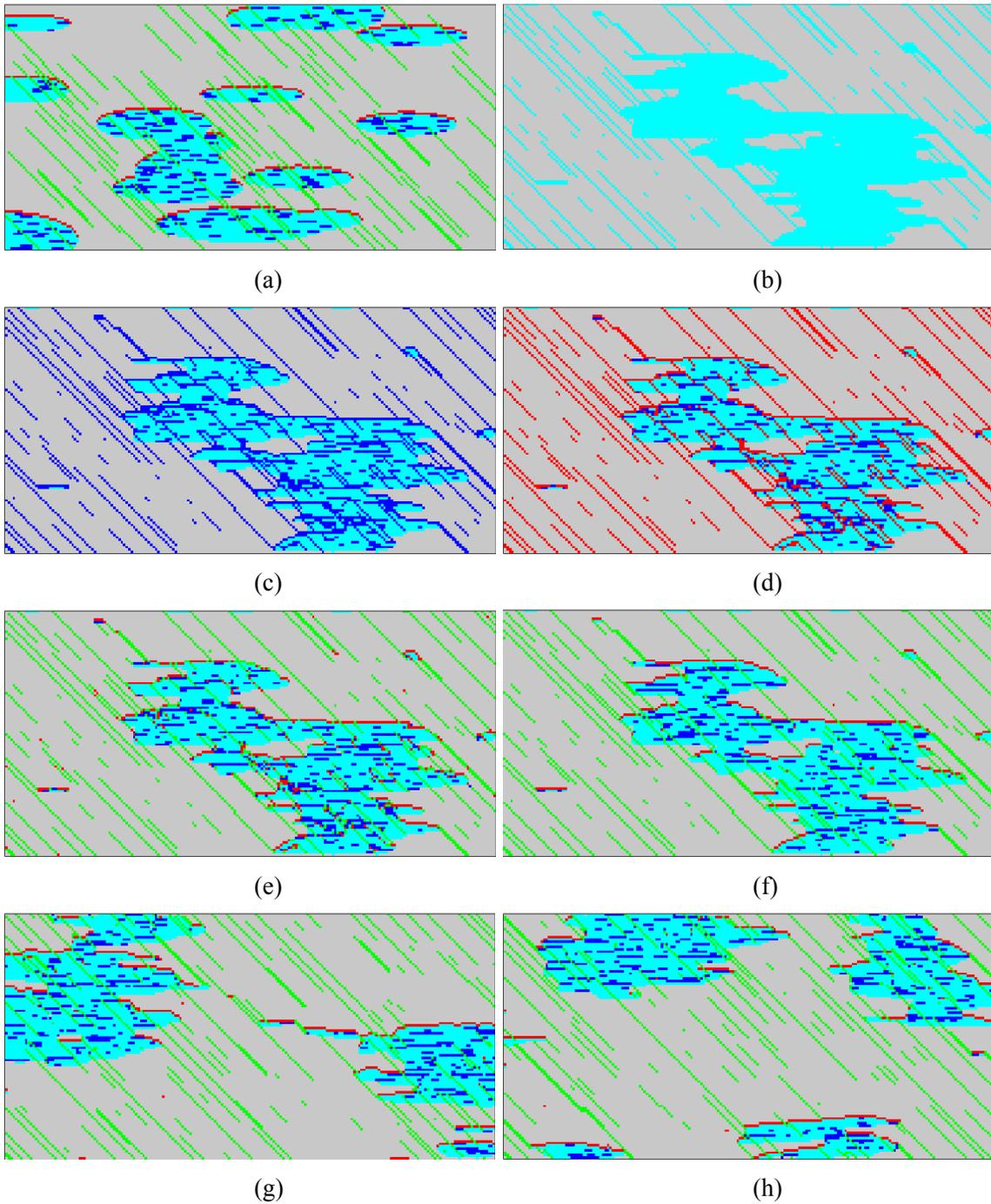


Figure 3: A stepwise simulation. (a) Training image [1]. (b) First step: background rock code vs. all others. (c) Background frozen, teal facies simulated vs. remaining (blue, red, green). (d) Teal frozen, blue simulated vs. red/green combined. (e) Blue frozen, red/green simulated on remaining unfrozen nodes. (f) Image cleaned, leaving only background frozen. (g) and (h) Two more finished realizations using the same method.

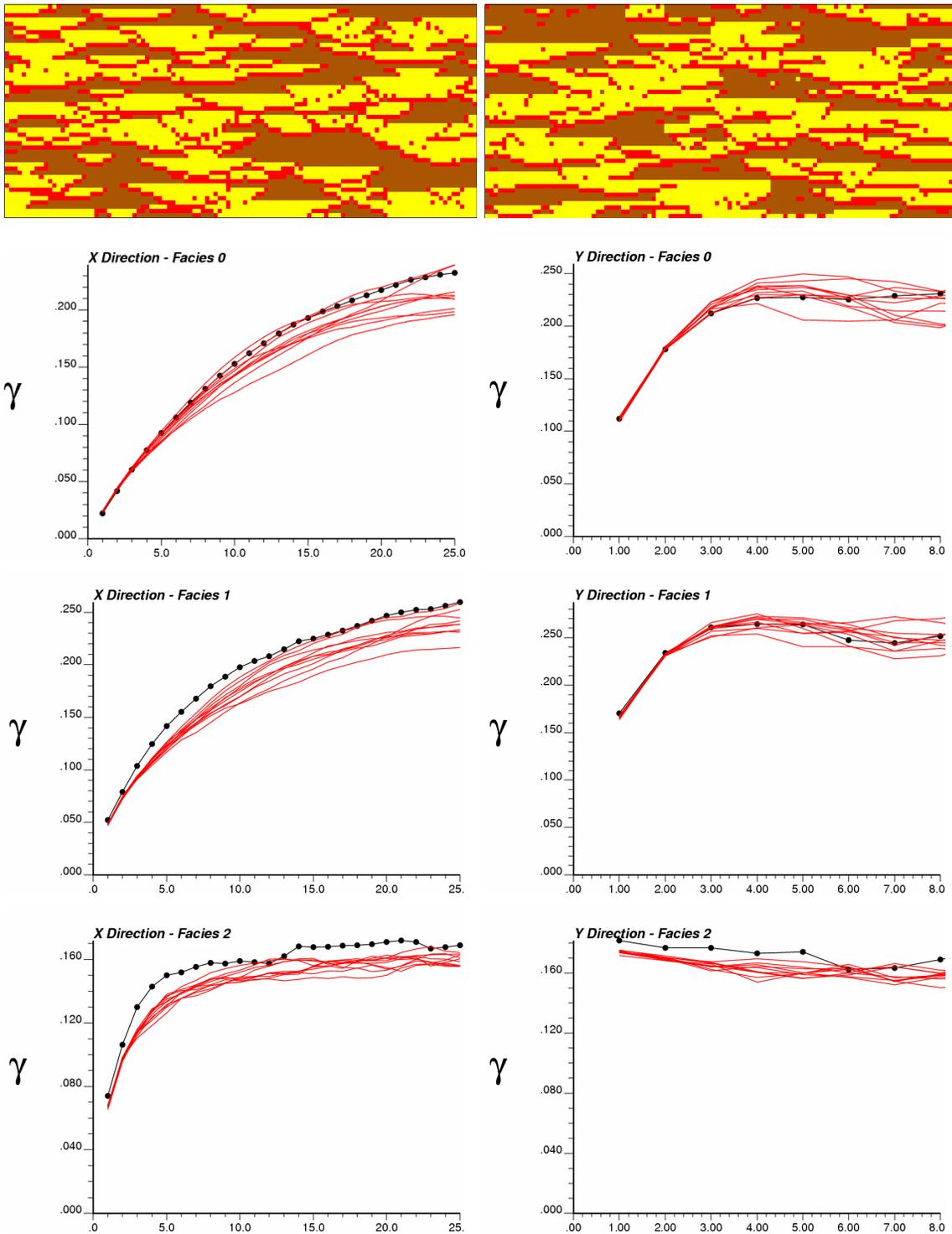


Figure 4: Top left: A TI from FLUVSIM featuring numerous channels. Top right: An MPASIM realization deriving the MP statistics from the TI. Lower: indicator variograms of the TI and 10 realizations, for each of the three facies in both the X and Y directions. Facies 0 is the brown (background) colour, facies 1 is the yellow channels and facies 2 is the red.

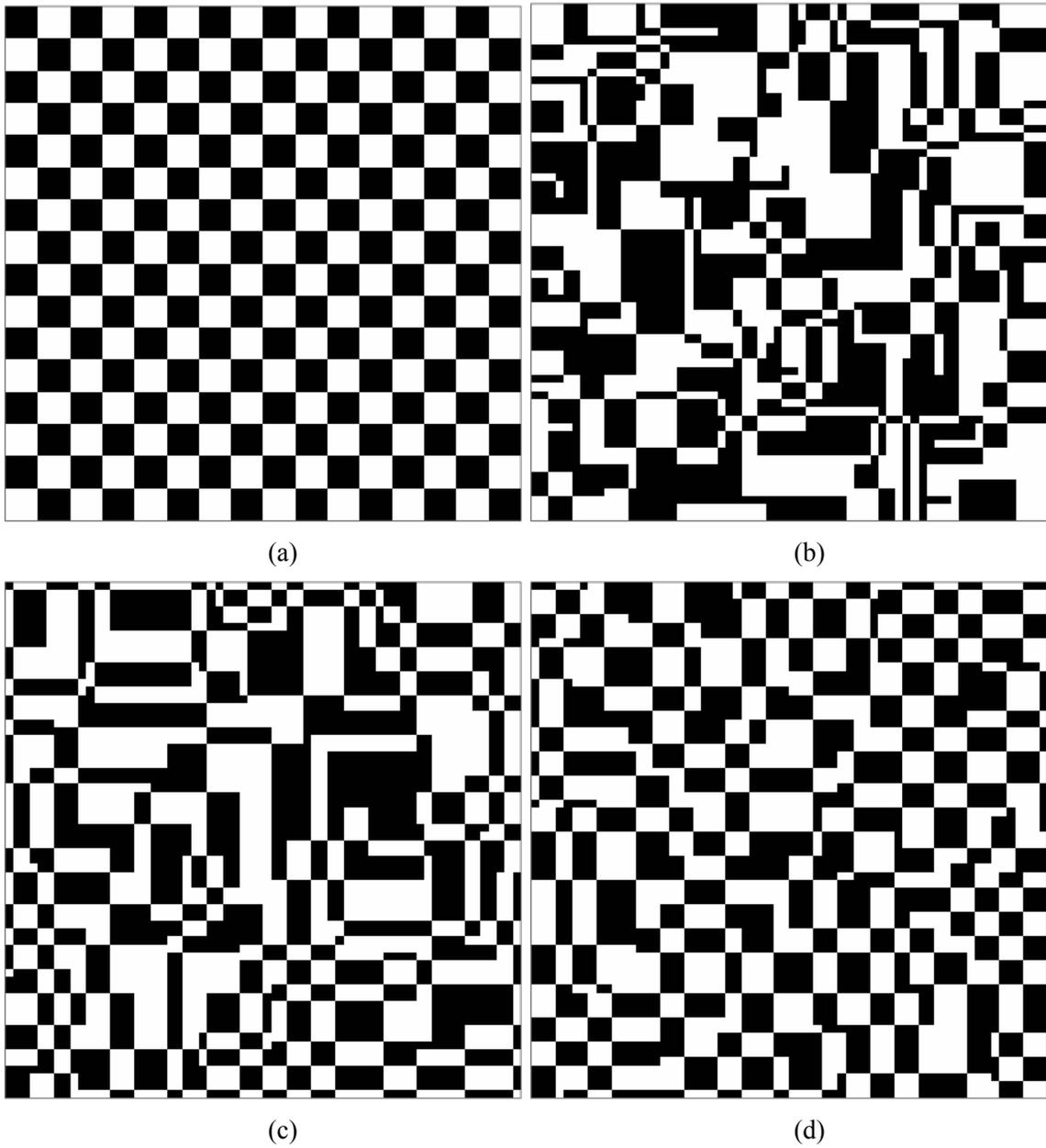
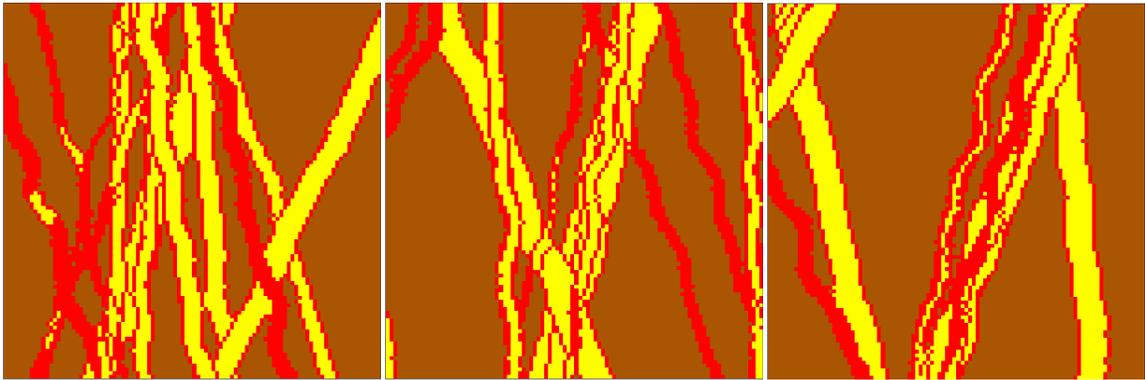
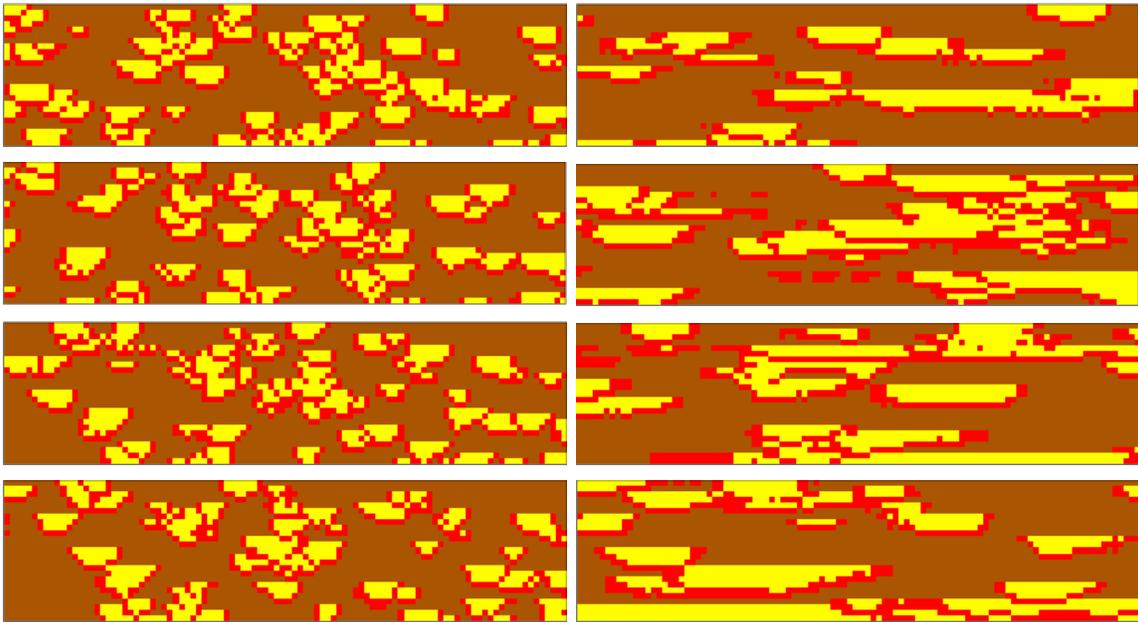


Figure 5: (a) A 64x64 pixel TI with 4x4 squares. (b) An MPASIM realization using a 2x2 MP pattern. (c) An MPASIM realization using a 3x3 MP pattern. (d) An MPASIM realization using a 4x4 MP pattern.



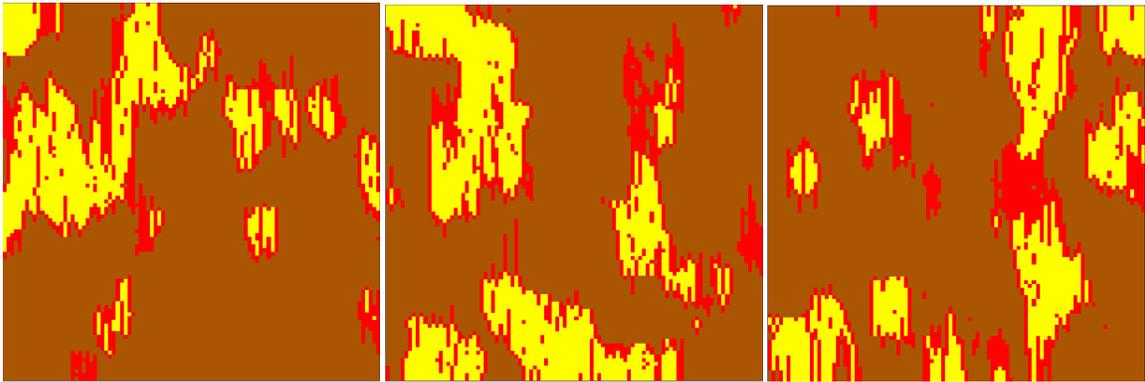
(a)



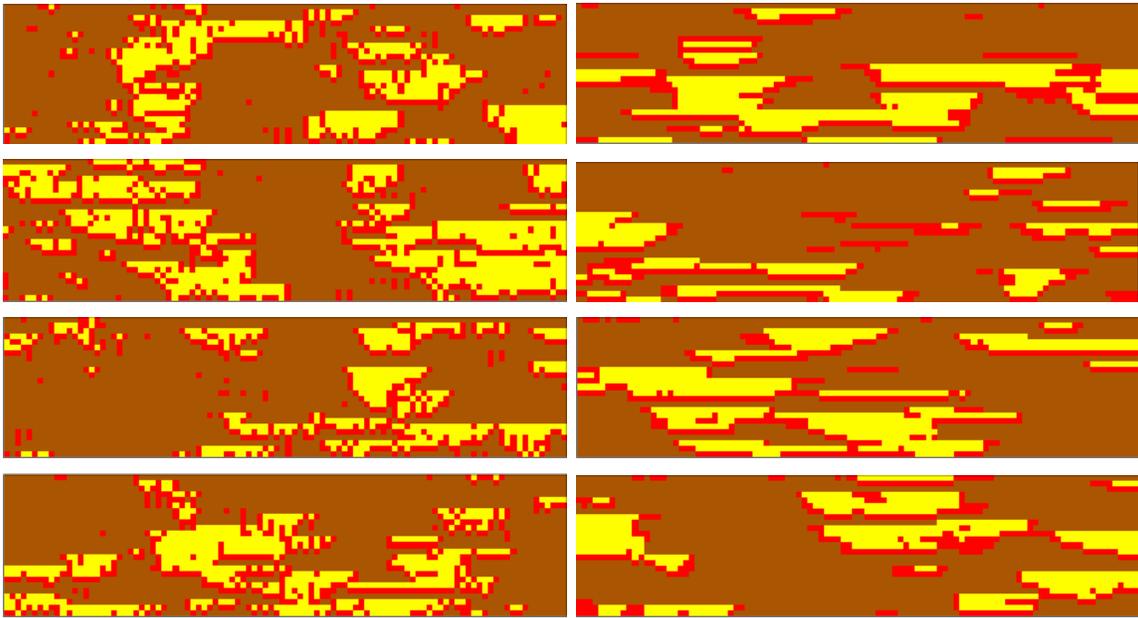
(b)

(c)

Figure 6: Sections of a 3D FLUVSIM TI. The size of the image is 100x100x25 nodes. There are three rock codes, brown (zero), yellow (one), and red (two). (a) XY; (b) XZ; (c) YZ.



(a)



(b)

(c)

Figure 7: Sections of an MPASIM realization generated using the TI in Figure 6 with a 2x2x2 MP pattern. (a) XY; (b) XZ; (c) YZ.

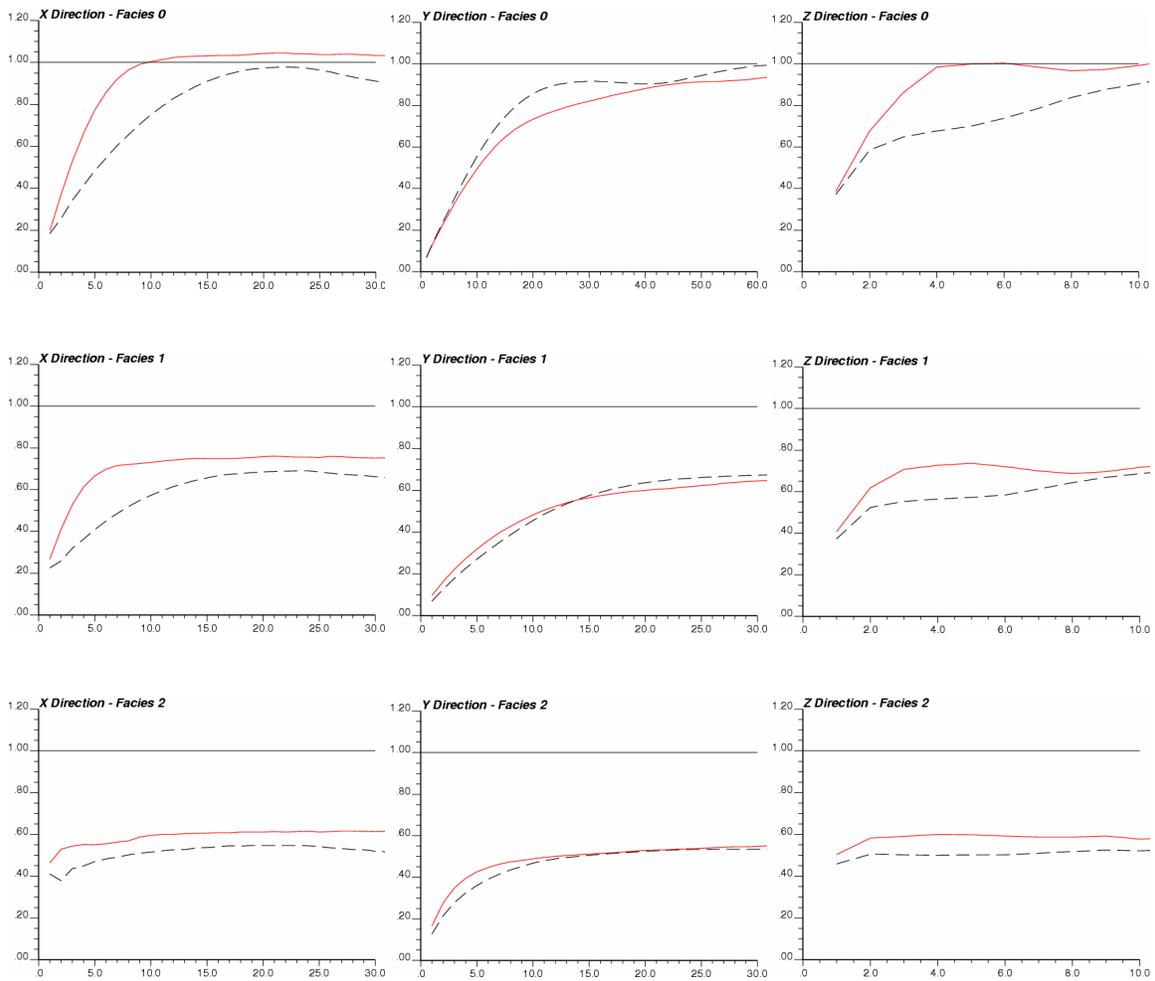


Figure 8: Indicator variograms for the FLUVSIM TI shown in Figure 6 and MPASIM realization in Figure 7. The rows are: Top, facies 0; Middle, facies 1; Bottom, facies 2. The columns are: Left, X direction; Middle, Y direction; Right, Z direction. The red lines are the TI variograms and the black dashed lines are for the simulation.

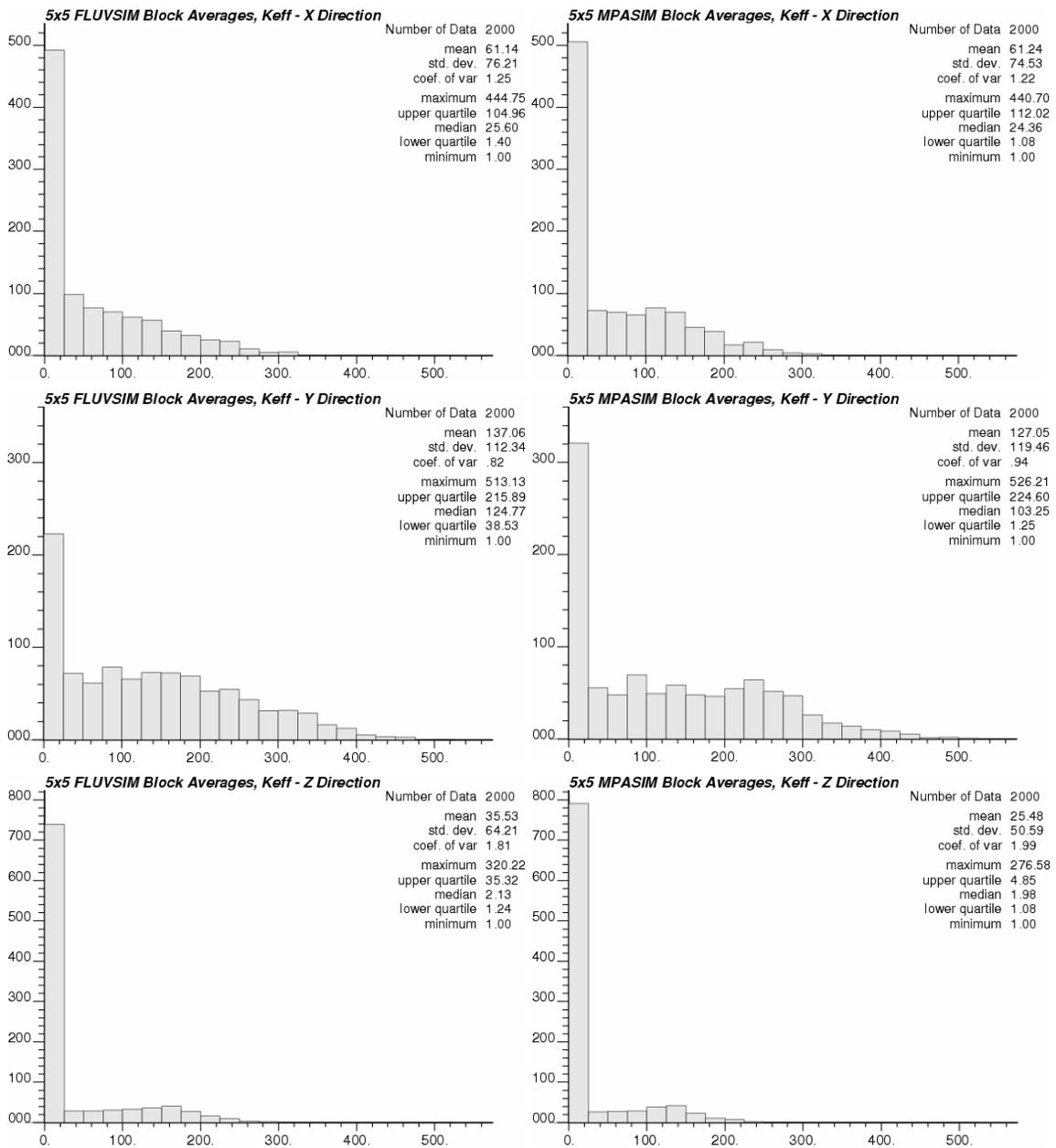


Figure 9: Histograms of effective permeabilities of 5x5 blocks for the FLUVSIM TI shown in Figure 6 (left) and the MPASIM simulation shown in Figure 7 (right). Top, X direction; Middle, Y direction; Bottom, Z direction.