Enhancing multiple-point geostatistical modeling: 2. Iterative simulation and multiple distance function

Pejman Tahmasebi¹ and Muhammad Sahimi¹

¹Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, California, USA

Abstract This series addresses a fundamental issue in multiple-point statistical (MPS) simulation for generation of realizations of large-scale porous media. Past methods suffer from the fact that they generate discontinuities and patchiness in the realizations that, in turn, affect their flow and transport properties. Part I of this series addressed certain aspects of this fundamental issue, and proposed two ways of improving of one such MPS method, namely, the cross correlation-based simulation (CCSIM) method that was proposed by the authors. In the present paper, a new algorithm is proposed to further improve the quality of the realizations. The method utilizes the realizations generated by the algorithm introduced in Part I, iteratively removes any possible remaining discontinuities in them, and addresses the problem with honoring hard (quantitative) data, using an error map. The map represents the differences between the patterns in the training image (TI) and the current iteration of a realization. The resulting iterative CCSIM—the iCCSIM algorithm—utilizes a random path and the error map to identify the locations in the current realization in the iteration process that need further "repairing," that is, those locations at which discontinuities may still exist. The computational time of the new iterative algorithm is considerably lower than one in which every cell of the simulation grid is visited in order to repair the discontinuities. Furthermore, several efficient distance functions are introduced by which one extracts effectively key information from the TIs. To increase the quality of the realizations and extracting the maximum amount of information from the TIs, the distance functions can be used simultaneously. The performance of the iCCSIM algorithm is studied using very complex 2-D and 3-D examples, including those that are process-based. Comparison is made between the quality and accuracy of the results with those generated by the original CCSIM algorithm, which demonstrates the superior performance of the iCCSIM.

1. Introduction

This paper represents Part II of a series devoted to improving the accuracy and efficiency of a pattern-based simulation method that we recently proposed, the cross correlation-based simulation (CCSIM) algorithm [Tahmasebi et al., 2012; Tahmasebi and Sahimi, 2012, 2013; Tahmasebi et al., 2014], for generating realizations of large-scale porous media (LSPM). The method uses a training image (TI), i.e., an image or a conceptual framework that contains some essential features of the spatial heterogeneity and continuity of the LSPM. It then utilizes a cross-correlation function (CCF) for matching the realizations with the TI. A computational grid is used in which the neighboring grid blocks or templates share overlap (OL) regions, which are used to match the patterns in the realizations with the TI, and make the transition from one grid block seamless. We refer the interested reader to our original papers and references, as well as Part I, for more details.

Part I of this series addressed some of the difficulties that one may encounter when implementing the CCSIM algorithm. Similar to any other method in this research area, the CCSIM method is also sensitive to the specifications of the patterns of heterogeneities in the TIs, such as the boundaries and the number of similar patterns. Part I reconsidered the original CCSIM and proposed two significant improvements for accurately reproducing large-scale patterns of heterogeneities in porous media. One was an effective boundary-correction method based on the graph theory by which one identifies the optimal cutting path in two-dimensional (2-D) porous media (cutting surface in 3-D) for removing the patchiness and
discontinuities in the realizations of a porous medium. The second improvement was a new pattern adjust-
ment method that automatically transfers the features in a pattern to one that seamlessly matches the sur-
rounding patterns. Part I combined the original CCSIM algorithm with the two methods and tested the new
algorithm using various complex 2-D and 3-D examples.

Before we begin to discuss the purpose of this paper, we should emphasize that, although the methods
that we proposed in Part I were tested only with the CCSIM algorithm, they are, in fact, quite general and
applicable to other pattern-based geostatistical simulation methods.

The TIs, which are stochastic in nature, have been shown to be very promising alternative sources of infor-
mation for stochastic simulation, because they provide a natural framework for developing models of the
LSPM that include higher-order statistical properties of the media, replacing the two-point covariance-based
information. But, they also have shortcomings that were described in Part I, as well as in our earlier papers
[Tahmasebi et al., 2012]. Briefly, in addition to the issues that we took up in Part I, the current problems
regarding the quality of stochastic TIs may be classified into the following groups:

1. **Conditioning the model to dense point/secondary data sets** (e.g., well and seismic data): Generally speaking,
the issue is a main reason that pattern-based methods produce low-quality realizations of porous media,
because various distance functions, i.e., the functions that measure the similarities and dissimilarities
(hence the name distance) between the realizations and the data, must be evaluated simultaneously. The
most important of such distance functions is one between the patterns of heterogeneities in the TI and
the model that is used in all unconditional/conditional simulations. Another of such distance functions is
one between the conditioning data that must be honored exactly in conditional simulation. Because not
all the data have the same significance, one must assign proper weight to each data set in order to select
a pattern that preserves the continuity and the constraints imposed by the conditioning data.

2. **Lack of similarity between the internal patterns in the TI**: The generation of a comprehensive and represen-
tative TI is very challenging. Therefore, one should expect significant discontinuities and patchiness in
the pattern-based techniques in conditional and even unconditional simulations. Due to requiring inten-
sive computations, however, providing a very large and rich TI and utilizing it in the simulation are costly
for most of such methods.

3. **Deficiency of the current similarity functions for quantitative modeling**: Most of the available distance func-
tions belong to single-point distance group, i.e., they are not accurate and select mainly a pattern that is
different from what an expert may decide to use. Such distance functions as variogram and the Euclidean
distance convey only a limited amount of information. Intuitively, a set of more informative distance func-
tions can improve the pattern reproduction.

In order to have better understanding of the current issues and the goal of the present paper, all the afore-
mentioned problems are summarized in Figure 1, where a complex continuous TI of a porosity field is
shown in which reproducing the long-range connected channels is, of course, important. A number of well
logs and seismic data set are also given in Figures 1b and 1c, respectively. Such conditioning data make the
simulation very difficult. Clearly, three distance functions for matching the patterns in a realization of the
porous medium, on the one hand, and those of the TI representing the porosity field, the well logs, and
seismic image, on the other hand, must be calculated. Therefore, any realization to be generated must deal with a considerable number of patchiness and discontinuities caused by the three different sets of data. One such realization is shown in Figure 1d. Undoubtedly, due to the discontinuities in the realization, it is not appropriate for flow modeling and cannot be physically justified geologically. The long-range connected channels, however, can be seen in the realization shown in Figure 1e, generated by a new iterative CCSIM (iCCSIM) algorithm to be described in this paper, which is free of any patchiness. The well logs and the general variability depicted by the seismic data are also reproduced correctly in this realization. It also should be noted that, for the sake of illustration, the location of hard data and also seismic data distribution are selected in such a way that they produce significant amount of patchiness.

Although the method proposed in Part I addressed certain aspects of matching patterns and the discontinuities on the boundaries between the OL regions, it must still be improved. To see this, consider Figure 1d, which indicates that the realization shown there still contains some discontinuities. This is due to the relatively large amount of the hard data that must be honored exactly, and the existence of some other auxiliary data that must be incorporated, which give rise to multivariate geostatistical simulation [Tahmasebi and Sahimi, 2015a]. Therefore, in this paper, the improved CCSIM algorithm developed in Part I is further refined in order to enhance the quality of the stochastic realizations that it produces. This is done by using an error map that provides the locations of the patchiness and discontinuities in the initial realization. Then, leveraging the graph theory in the CCSIM described in Part I, the error map is used to guide the algorithm to “repair” the patchiness and discontinuities. Along the repairing step, a multiple-distance function is also introduced that improves the pattern matching significantly.

The plan of this paper is as follows. In the next section, we provide an overview of the problem that we address in this paper. Section three introduces the iterative CCSIM and describes the algorithm in detail. The multiple distance functions and their use are introduced and discussed in section 4. The accuracy of the new algorithms is tested in section 5 using several complex TIs. The paper and the series are summarized in section 6.

2. Patchiness and Discontinuities

Generally speaking, patchiness, i.e., sharp discontinuities between various patterns, exists in most of the pattern-based higher-order geostatistical methods that, as discussed, are mainly due to conditioning of the realizations to dense data and the lack of the availability of a rich, representative TI. The issue is more visible in the algorithms that use a raster path [Daly, 2004], i.e., the path along which a realization is constructed block by block. In other words, unlike the random-path algorithms, the simulation points and grid blocks are visited in a structured manner such that, starting from a corner of the grid, they are simulated sequentially along a raster. In addition to the aforementioned issues, however, it is useful to first discuss why raster-path algorithms are still very appealing.

2.1. Raster Versus Random Paths

For over a decade, algorithms that utilize a raster path have been popular in the stochastic modeling methods. Daly [2004] was probably the first to present a Monte Carlo algorithm that proceeded along a raster path and produced acceptable results. Then, El Ouassini et al. [2008] implemented a patch-based method together with a raster path. Next, Parra and Ortiz [2011] used a similar path in their study. Finally, Tahmasebi et al. [2012, 2014] and Tahmasebi and Sahimi [2012, 2013] used a raster path in the CCSIM algorithm and achieved high-quality realizations in a matter of a few CPU seconds. A common character of all the available raster-path algorithms is their ability for generating high-quality patterns that can hardly be obtained by those methods that utilize random paths.

The philosophy behind such methods, and particularly those that use a raster path, is the low number of constraints, enabling them to produce high-quality realizations. For example, in 2-D simulations using the CCSIM algorithm, the number of overlaps (OLs) between the neighboring blocks (see Part I and our earlier papers), which represent constraints that one must match along the raster path varies from one (on the external boundaries of the computational grid) to two (in the interior regions). However, the number of the OLs can fluctuate from one to four in the algorithms that use random paths. As a result, by increasing the number of the OLs one should expect more patchiness and discontinuities. In other words, developing
a pattern that matches the previously simulated patterns, each of which is from different parts of the TI, is very difficult, if not impossible. Therefore, decreasing the number of the OLS is very helpful, and can help generation of a high-quality realization. To this end, regular (nonrandom) raster paths offer the best possible way of reducing the number of the OLS.

One of the main differences between image construction methods and stochastic simulation in the earth sciences is, however, dealing with conditioning data (e.g., point and secondary data). As one of the foremost drawbacks, raster paths cannot account for the conditioning data that are ahead of them, i.e., the condition data at location that have not been visited yet by the path. Therefore, one should expect a bias toward the conditioning points. To avoid this difficulty and bias, Tahmasebi et al. [2014] suggested a cotemplate method that accounts for the data that are ahead of the simulation path. In this paper, another complementary method, based on the traditional random-path geostatistical methods, is presented that has the advantages of the raster-path methods while using a random simulation path. Note, however, that we do not abandon the use of a raster path. The initial realizations are still generated by using raster paths, but, then, random paths are used in order to improve their quality and conditioning; see below. Furthermore, another issue with the current pattern-based stochastic methods is their reliance on a sole Euclidean distance, which has been proven to identify patterns in the TI that are visually not similar to those in the realization [Lowe, 1999]. Therefore, in this study, the idea of using multiple distance functions that try to identify higher-order differential elements will be discussed.

3. Iterative CCSIM

As discussed, various factors, such as the raster path, conditioning to dense point/secondary data, the finite size of the TI, and other algorithmic shortcomings cause various discontinuities, which in turn affect the flow modeling and reservoir/aquifer planning. Therefore, the necessity of preventing such discontinuities is very important. It is of course clear that a geologically realistic model is developed when one can generate a realization of the porous medium as close to the TI as possible. For example, geomechanical data and concepts can be integrated with the current stochastic modeling methods, which generate more realistic realizations.

One solution to avoiding the discontinuities and other conditioning problems was suggested by Tahmasebi et al. [2012], which is to use another raster path in a different direction to improve the quality of the realizations. The idea works very well when one varies the template (grid block) size. Using different template sizes, one achieves two goals simultaneously: correcting the realizations that are based on an iterative process that improves them, and accounting for the multiscale features of the porous medium of interest. For example, by changing the template size hierarchically, the large and small scale structures can be simulated. Clearly, starting from a large template, the main patterns in the TI can be mimicked. Then, one iterates the realization by using smaller templates that enable the user to include the fine-sale features of the porous medium in the model. The use of smaller templates can continue iteratively until a satisfactory match with the TI is obtained. The idea is not, however, feasible for 3-D or very large 2-D simulations. First, repeating the initial simulation on the already generated realization is computationally very demanding due to the use of the entire data events (the patterns of heterogeneities in the OL regions) in the calculations with the CCF or the distance function. Second, one needs to check a few realizations to identify the number of iterations that are needed, which is also time consuming. Third, using the previous simulation path does not allow the algorithm to account for the long-range connectivity between the hard data, making them not useful. Together, such algorithms make one version of what we refer to as the iterative CCSIM (ICCSIM) algorithm. In this paper, these issues are addressed.

An alternative to alleviate the computational burden of the above iterative concept is to use an auxiliary map that identifies the locations in the realization that must be “repaired,” i.e., where the discontinuities must be removed. We call this the error map. The map can simply be constructed for, for example, continuous TIs using a measure of dissimilarity between the OL regions, previously simulated, and the newly-selected patterns, i.e., the distance function that evaluates the dissimilarities between the already generated realization and the selected heterogeneity pattern from the TI. For example, if the measure is based on a Euclidean distance, it is defined precisely the way the distance between two points in space, \( x = (x_1, x_2, x_3) \) and \( y = (y_1, y_2, y_3) \), is defined, namely, \( d^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 \), except that instead of the
coordinates we use the pixel values. The mismatch with the hard data can be added to the map later on. Intuitively, the Manhattan distance (a binary distance defined to be zero for similar patterns and 1 otherwise) can be used for categorical (discrete) TIs. Then, one constructs a path for repairing those regions indicated by the error map. Next, the remaining errors in the OL regions are eliminated using a semipixel-based version of the CCSIM (see below). For example, Figure 2 represents the error map for a conditional simulation using the TI and the secondary data shown in Figure 1. The map indicates the locations where the errors (mismatches) occurred, hence needing repair. Therefore, instead of resimulating the entire realization once again, one concentrates on repairing the high-error areas. Using the error map, particularly for 3-D simulation, reduces the computation time significantly. So, the error map acts as the input data for the iCCSIM.

To carry out the repairing, we use one advantage of the graph-based CCSIM [Kwatra et al. 2003], developed in Part I, which accounts for the old cutting paths, i.e., the path along which the patterns in the TI and those in the realizations were cut in order to generate two matching segments next to each other. Suppose that several patterns already exist in the midst of the simulation model, and that we wish to place a new pattern in a given region. Clearly, laying down a new pattern of heterogeneity without any consideration for the already existing structure will create a noticeable discontinuity in the realization. Similarly, the costs (errors) of the earlier cutting paths are also available when the graph-cut problem is solved. The old costs can be combined with the new graph-cut problem to determine the new optimal cutting path. This is illustrated in Figure 3, where all the previously simulated patterns in the realization are represented by $\mathbf{P}$, and the newly selected candidate pattern is denoted by $\mathbf{R}$. The old cutting paths can now be represented as new nodes $S_i$ between the regular cells. Then, each new node $S_i$ is connected to the new pattern using an arc. The cost of the arc is the same as the previously saved cost, $C(s, t, R, P)$, where $s$ and $t$ are adjacent cells for the cutting path. For example, in Figure 3 an old cutting path is located between cells 5 and 9. Therefore, a new node $S_1$ is inserted in the space between the two. Using an arc of the old cutting path with the cost $C(5, 9, R, R) = S_1$ also connects to the

---

**Figure 2.** The error map for the OL regions in a conditional simulation using the TI shown in Figure 1a. The error map is normalized to [0 1]. The blue regions represent low errors that do not need to be resimulated or repaired.

**Figure 3.** (left) Accounting for the old cuts for identifying the best cutting path. (right) Graph formulation between the old and new cuts.
newly selected pattern $R$. The other arcs from, for example, $S_1$ to cell 9 and from 5 to $S_1$, are marked as $C(5, 9, R, P_9)$ and $C(5, 9, P_5, R)$.

Consequently, in this case three scenarios are possible. (i) If the arc between the nodes in the old cutting path and the new pattern is cut, it means that the old cutting path remains in the realization. (ii) If the arc is not cut, then the old cutting path will be covered by the new pattern and will not be used any more. (iii) One of the arcs between the nodes of the old cutting path and the cell belongs to the current cutting path, meaning that a new cutting path is generated and added to the already existing path. For example, in Figure 3 the final new cutting path is shown by a black-dashed line. Nodes $S_4$ and $S_5$ from the old cutting path are removed. $S_4$ is replaced by the new cutting path, and a new cutting path is generated between cells 8 and 6, 10 and 11, and 9 and 13.

The extended graph formulation of Figure 3 is used as a tool for addressing the problem that the present paper studies, and is shown in Figure 4. Suppose that several patterns have already been placed in the left realization and, based on the iCCSIM, we wish to remove the patchiness from the upper right corner (black light box). First, a pattern using the CCSIM algorithm with the given template and OL size is selected (Figure 4, middle). Then, the cells on the border of the location where the new pattern should be inserted are considered as hard data or constraints that must be honored exactly. In other words, the edge cells are the same as those in the realization. Such constraints are shown using the gray arcs from the boundaries to the realization (Figure 4, right). Furthermore, one can even force any cell to be copied from the new selected pattern. In this example, nodes 7 and 10 are connected to the selected new pattern using two arcs. A more illustrative example for this application is shown in Figure 5.

There is yet another issue that should be addressed. The error map of a realization generated by the iCCSIM can produce artifacts and some discontinuities. In other words, although by using the error map one cannot access the acceptable pixels in the realization, most of the used pixels in the selected data event come from the inaccurate parts of the realization. Therefore, to ensure that the data event includes accurate pixels, a new dilated error map using a square of size of 4 is generated. Two examples of the generated dilated error map are shown in Figure 6. As displayed there, the new generated error maps contain more surrounding pixels and, thus, the data event will not be limited to the invalid pixels.

We now have all the necessary inputs and algorithm to begin the simulation using the iCCSIM. All the errors are normalized and, thus, are between zero and 1. We also set a threshold for the errors. If they are smaller than the threshold, we accept them as accurate enough, but if they exceed it, we repair them. The threshold used in this work, based on some preliminary simulation, is 0.2. Then, a region that needs repairing/enhancement (according to the threshold) is selected using the error map. The best candidate pattern is inserted in the selected point, given the data event whose cells (pixels or voxels) are fully specified. Then, the error for the newly selected pattern is computed, and added to the error map to update it. Next, the location with the highest error is selected as the next visiting point for repair in the random path. For
categorical (discrete) TIs, however, the error map is also not continuous. In this case, all the mismatched points in the visiting path are considered, but the locations where the mismatches are less than the threshold 0.2 of template size are ignored. One can use a smaller $\epsilon$, which increases the computation time. The idea is summarized in Figure 7. As can be seen, the iCCSIM starts with a realization in which high mismatch with the hard data has produced patchiness. The iCCSIM first uses a large template size and gradually reduces it. The patchiness is removed partially using the graph-cut method (see Part I). Proceeding with the iterative algorithm, the discontinuities and mismatches are removed further. For example, the patchiness in the second and third iterations is removed, and at the same time all the hard data are honored. Further iteration on the previous realizations, for example the sixth iteration, removes all the existing discontinuities.

Figure 5. Repairing the upper right part of (b) of a realization shown in Figure 5a. First, a matching candidate pattern using the given template size (red box in Figure 5b) is selected, which is shown in Figure 5c. Then, using the graph-cut formulation, the best cut is identified; see Figure 5d. The realization is overwritten using the resulting cutting path, and shown in Figures 5e and 5d. The vivid yellow box in Figure 5b indicates the locations that are used as the internal constraint.

Figure 6. The effect of using dilation on continuous (a) and binary (c) error maps. The results are shown in Figures 6b and 6d, respectively.
should be noted that due to elimination of most of the patchiness and mismatches in the second iteration, there is no need to use more iterations, as no major geological alteration can be made.

Note that, although in Figure 7 after three iterations we have obtained a realization without any mismatch with the hard data, we have continued the process for two more iterations because of minor discontinuities in the realization. Naturally, one can stop the iteration process at any stage that is deemed suitable. Given the uncertainties in the data and the unknowns for any large-scale porous medium, one does not have to further iterate the realizations to remove minor discontinuities, but the method allows one to do so, if desired. We find that the choice of the distance function is important, as some of them lead to a poor pattern reproduction. In the next section, this important topic is discussed further, and a solution is suggested.

4. Multiple Distance Functions

The foundation of pattern-based techniques relies mostly on the distance function that controls the quality of stochastic modeling. Using very simple distance functions may generate a considerable number of discontinuities. Thus, as discussed, the poor quality of a model is not only due to having a poor TI, but may also be due to the type of the distance function used. Use of a very large TI in which a specific heterogeneity pattern is repeated in various locations and is surrounded by other distinct patterns is not, however, practical as it is computationally very costly. Therefore, one solution is to rely on the available TIs and try to use better (dis)similarity measures—distance functions—to consider various aspects of the patterns’ variability. Using a single distance function for all the various TIs, including complex continuous and categorical TIs, is not, however, wise. As a solution, in order to have a more robust procedure, one needs to use various forms of distance functions simultaneously. This would be similar to optical devices, each of which is
applicable to a specific wavelength and features. In the present context, various distance functions can reveal different aspects of the complexities of a TI. The Euclidean distance accounts for single-point similarity, whereas the same distance function cannot account for multiple-point information. Eventually, by combining various distance functions, one compensates the weakness of each. In what follows, various distance functions and their application to higher-order geostatistical methods are described. We describe them for 2-D media. Their extension to 3-D systems is straightforward.

4.1. Distance Transform

Due to the high complexity of most of the TIs, it is preferred to infer additional information about the values of a property around a point. At the same time, converting a simple TI to a more informative multiple-point image is very useful. The distance transform (DT) extracts such information from a binary TI [Barrow et al., 1977; Borgefors, 1988]. It uses a distance metric, such as the Euclidean or quasi-Euclidean distance, to calculate the distance of a pixel with its nearest nonzero (i.e., object) pixels. Mathematically, for \( p \) as an arbitrary cell in a window of binary (uniform matrix plus channels) TI and a closest set of \( W \), the DT is defined as:

\[
DT(p, W) = \min_{t \in W} (\varphi(p, t)),
\]

where \( \varphi(p, t) \) is the Euclidean distance between \( p \) and \( t \). Therefore, the output image contains zero for the locations defining the object—the channels in Figure 8—but also contains nonzero as one gets farther away from the objects. This concept is illustrated in Figure 8.

As mentioned, this method can be used for binary and multifacies TIs. For example, the latter type of a TI needs to be first discretized into a binary image and then the distance transform method can be applied. The method provides an excellent overview of the available features in the TI, and more information about the complexity of the TI.

4.2. Gradient-Based Methods

The gradient of a TI indicates directional variations that provide significant amount of information in various directions. A simple distance function based on the Euclidean distance does not account for the frequency content of the TI. Patchiness occurs mostly by the low-frequency regions of the TI. In other words, due to their lack of the similarity with other regions in the TI, patterns with low frequency of replicates usually generate discontinuity in the simulation grid. Therefore, the frequency content can be integrated using a gradient map of the TI, described below. The gradient function provides mostly two pieces of the information: the direction of the gradient that indicates the largest possible variation, and the length that corresponds to the rate of variation.

For this aim, the gradient, or intensity of the TI is calculated at each point. This yields a new image—the gradient image—with discrete points that represent the intensity at various locations that cannot be used because it represents a discrete distribution, whereas one must use a continuous distribution. One can assume that the discrete points are a sample of a continuous intensity function. The simplest and most
effective method is to form the convolution of the TI and a kernel. In this paper, the Sobel gradient operator is used [Gonzalez, 2009] as the kernel. The operator uses two $3 \times 3$ kernels that are convolved with the TI to calculate approximations of the derivatives (gradients)—one for horizontal changes, and one for vertical. The horizontal $G_x$ and vertical derivative approximations $G_y$ are defined as follow:

$$
\begin{align*}
G_x &= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ -1 & 0 & 1 \end{bmatrix} \ast \text{TI}, \\
G_y &= \begin{bmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix} \ast \text{TI},
\end{align*}
$$

(2)

(3)

Where $\ast$ donates the 2-D convolution operation. Finally, the quantity

$$
G = \sqrt{G_x^2 + G_y^2},
$$

(4)

is calculated that represents the magnitude of the gradient. Note that both the distance transform and the gradient-based methods are stand-alone techniques. That is, one can implement one or both methods (or none for that matter) in order to improve the quality of the realization. The decision as to whether one should use the two methods depends on the complexity of the TI and the precision of the realizations that one requires.

By implementing the resulting function, one distinguishes better between two seemingly similar heterogeneity patterns that, in fact, have different structures. In other words, a simple Euclidean distance fails to identify a match between two patterns with globally similar structures and features that may, however, contain significant but seemingly minor differences. Therefore, by extracting the main features, the image gradient identifies significant differences between two seemingly similar patterns, and then reduces the matching error.

An example that presents the result of integrating the Sobel gradient image is presented in Figure 9. The integration of a simple TI with useful information that is provided by its gradient image improves the modeling significantly. It should be noted that the gradient image is very useful when one deals with highly structured TIs.

There are also some other distance functions, such as scale-invariant feature transform [Lowe, 1999] and the Gabor filters [Gabor, 1946] that we have experimented with, together with the CCSIM. But, our study indicated that they cannot provide high-quality realizations. For example, by using the Gabor filter one extracts various textural data for each Gaussian kernel function. Moreover, we should keep in mind an important issue concerning the use of the Gabor filter, namely, it’s very high computation time. The Gabor filter is a collection of several different filters, use of each of which results in a separate feature map. The feature maps need to be treated as multiple TIs, first used by Tahmasebi and Sahimi [2015a]. Others [Liu et al, 2005] used multiple TIs in the past, except that they used them in order to evaluate the uncertainty associated with geological processes. Thus, using Gabor filters offers a trade-off between the amount of extracted information and the computation time. Even if a collection of all the filters is used, generating a high-quality realization is not guaranteed.

After deciding on the type/number of the distance functions, one integrates them all. Then, a set of distance functions consisting of all of the available generated patterns, $P_f (P_{re} \subset F_{re})$ of the feature images, $F = \{F_1, F_2, \ldots, F_n\}$, using their corresponding function is utilized to define the overall distance for the pattern in a visiting point $P^n_{re} (P^n_{re} \subset \text{TI})$. Such distances are denoted by $M = \{M_1, M_2, \ldots, M_n\}$. For example, the similarity for two distance functions of value (the default distance in the CCSIM, $F_1$) and the DT ($F_2$) is expressed as follow:

$$
M_1 = \sum \lVert P^{c1}_{ii} (i, j) - P^{c1}_{i,j} (i, j) \rVert,
$$

(5)

$$
M_2 = \sum \lVert P^{c2}_{ii} (i, j) - P^{c2}_{i,j} (i, j) \rVert,
$$

(6)

where $P^{c1}_{i,j}$ and $P^{c2}_{i,j}$ are the candidate patterns in the TI and the generated feature image location $(i,j)$ in the simulation grid, respectively, and $\lVert \cdot \rVert$ represents the Euclidean distance (see above). Then, the overall distance function, in its simplest form, is defined by,
where $0 < \omega_f < 1$ is the weight for the $f$th distance function $M_f$. Note that one can also normalize $M_f$ to $[0,1]$ and use an equal weight for all of the distance functions, unless any of them must be given a higher weight because it has more influence on producing a more accurate pattern. The selection of the weights is an interesting and important problem, representing a complex optimization problem that is currently under study.

5. Results

We now demonstrate the accuracy of the proposed iCCSIM method by using various complex TIs. In what follows, when we refer to the CCSIM, we mean the improved method developed in Part I. For this aim, process-based TIs are used extensively. All the simulations were carried out on a computer with a 2.66 GHz CPU and 4 GB RAM. In what follows, the examples and results are described and discussed.

5.1. Two-Dimensional Binary Channel Model

As the simplest example, a binary channel model (uniform matrix plus channels) is considered, which is shown in Figure 10a, while the data for two wells are shown in Figure 10b. The results of conditional simulation using both the CCSIM and the iCCSIM are also shown in Figure 10. When comparison is made between the conditional realizations generated by the CCSIM, Figures 10c and 10d, and those by the iCCSIM, Figures 10e and 10f, it becomes clear that the iCCSIM produces very accurate results, with most of the channels in the TI reproduced and connected correctly. The improvement by the iCCSIM is better illuminated by the conditional variance maps, Figures 10g and 10h, which indicate that the
iCCSIM manifests a more symmetric uncertainty space. A variance map contains the variances at every point of the realizations over a large number of them. In this case, we computed the variance map for 100 realizations.

Similar to Part I of this series, in order to quantitatively test the accuracy of the iCCSIM, the same quantitative comparisons used in Part I are also made in this study. Thus, we first computed the multiple-point connectivity (MPC) probability, $p(r)$. The results for the realizations shown in Figure 10 are shown in Figure 11. Clearly, the similarity between the MPC function of the TI and the realizations is improved when the iCCSIM is used to generate the realizations. The reason is that the algorithm further refines the realizations to match with the TI and, consequently, results in better pattern reproduction. Such improvement is further investigated using the analysis of the distance (ANODI) and the multidimensional-scale (MDS) analysis, described in Part I. The results are presented in Table 1.

Clearly, the uncertainty space has not changed, but the pattern reproduction score indicates significant improvement. These results are also seen in Figure 11c, where most of the iCCSIM realizations are distributed close to the TI, indicating close similarity between the patterns of the realizations and the TI.

Also computed was the effective permeability of the realizations generated by the iCCSIM and the TI. The permeabilities of shale and sand channels were assumed to be 1 md and 1000 md, respectively. The effective permeability of the TI is 223 md. The average computed effective permeability over 50 realizations is 220 md, very close to that of the TI.
5.2. Two-Dimensional Multifacies

Characterizing complex meandering channels is a difficult problem in hydrology, oil and gas reservoir, and earth science. Such structures play an important role in both groundwater and energy resources, and provide the main flow paths [Deutsch and Tran, 2002; Pyrcz et al., 2009; Sahimi, 2011]. Therefore, the connectivity of such highly permeable channels is very important. In the present example, the TI is a 2-D meandering system, along with some seismic data shown, respectively, in Figures 12a and 12b. The auxiliary (seismic) data help tracing the global distribution of the channels in the porous formation. Some seismic information for this example, which represents soft data, is shown in Figure 12c, for which the geological model is not available. Additionally, facies information for two wells in this area is also available and shown in Figure 12d. Fifty conditional realizations (without using the well data) were generated to verify the lack of bias by the proposed iCCSIM algorithm. Two of such realizations are shown in Figures 12e and 12f. The corresponding ensemble average and conditional variance maps are shown in Figures 12g and 12h, respectively. It is clear that the ensemble maps, except the one that used seismic information as auxiliary data, do not possess any particular structure. Then, 50 conditional realizations were generated by the iCCSIM using the well data, two of which are shown in Figures 12i and 12j. The iCCSIM repaired the patchiness and resulted in the highly connected channels. Likewise, the ensemble average and conditional variance maps are also provided for the conditional simulation in Figures 12k and 12l.

A more precise look at the iCCSIM’s procedure for the example in Figure 12 is presented in Figure 13. Consider the displayed image in Figure 13a as the initially generated realization. A discontinuity on the left side of the meandering channel is seen. Using the TI in

<table>
<thead>
<tr>
<th>Table 1. The ANODI Scores for the Variability of Within and Between Realization and Total Ratio for the Iterated CCSI M (iCCSIM) and the OCCSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space of uncertainty (between)</td>
</tr>
<tr>
<td>Pattern reproduction (within)</td>
</tr>
<tr>
<td>Ratio (between/within)</td>
</tr>
</tbody>
</table>

Figure 11. The calculated multipoint connectivity function \( p(r) \) for the realizations generated by Figure 11a the CCSIM and by Figure 11b the iCCSIM. Gray curves represent the realizations and the TI is shown by black. (c) The MDS-ANODI plot for demonstrating the variability in the realizations generated by the CCSIM (green dots) and by the iCCSIM (blue dots). The red dot indicates the TI.
Figure 12. Application of the iCCSIM to a TI with complex meandering channel, shown in Figure 12a, using the auxiliary data in Figure 12b and conditioned to seismic data in Figure 12c and well logs in Figure 12d. Two conditional realizations, (e) and (f), along with the ensemble average and conditional variance maps for 50 realizations using the seismic data only are shown in Figures 12g and 12h. The same results using both the seismic and well data, (i) and (j), in addition to their ensemble average and conditional variance maps for 50 realizations, are also presented in Figures 12k and 12l. Note that the scale-bars in Figures 12b, 12c, 12g, 12h, 12k, and 12l are the same.
Figure 13a, a matching pattern is identified and inserted in the patchy area. To do so, we first use the iCCSIM algorithm to select the candidate pattern and, then, the optimal cutting boundary is computed. The cutting area is shown in Figure 13b. Then, the selected pattern is cut and placed in the selected area based on its provided boundary.

Figure 13 demonstrates that the iCCSIM algorithm captures the geological features of the complex meandering channel and improves the quality of the given realization iteratively. The iterative procedure continues until the input realization is corrected and the patchiness is removed completely. In addition to removing the discontinuities and patchiness, however, the iterative algorithm may be used to enhance the conditioning as well. The reason is that the initial process-based model is difficult to be conditioned to well data, but by using the proposed algorithm the conditioning to the well data is achieved.

For this example, the MPC function $p(r)$ of the iCCSIM-generated realization is compared to that of the TI in order to check the long-range connectivity of the meandering channels; see Figure 14a. The results indicate very good agreement between the realizations and the TI. Further evidence for the high quality of the realizations is obtained by the ANODI analysis; see Figure 14b that indicates that the realizations are mostly distributed around the TI. Such a close cloud of realizations is manifestation of accurate pattern reproduction by the iCCSIM.

The average effective permeability of the iCCSIM-generated realizations was also compared with that of the TI. The permeability of the blue, orange, and background shale facies were assumed to be 850, 1000, and 10 md, respectively. The effective permeability of the TI is 202 md, while the average effective permeability over 50 realizations generated by the iCCSIM is 197.58 md. Once again, the computed permeabilities indicate very close agreement.

Figure 14. (a) The calculated MPC function $p(r)$ for the iCCSIM-generated realization of Figure 10. The gray curves represent the MPC function for the realizations, while that of the TI is shown by black. (b) The MDS-ANODI plot for uncertainty space of the realizations (gray) and its comparison with the TI (red).
5.3. Two-Dimensional Continuous Field

The performance of the iCCSIM with a continuous TI shown in Figure 15a is tested next. The complexity of the example is twofold: a continuous TI, and the clustered well data in Figure 15b. Using the information 100 conditional realizations were generated, three of which are shown in Figure 15c. All the well data are reproduced exactly, while the realizations are free of any patchiness or discontinuity, at least visually. We come back to this point shortly. For the sake of comparison, the realizations generated by the iCCSIM for two paths, one 1-D raster and one random are compared. The ensemble average and conditional variance maps for each path are depicted in Figures 15d and 15e, respectively. Likewise, the conditional variance maps of the realizations generated by the iCCSIM for the 1-D raster and random paths are shown in Figures 15f and 15g, accordingly (the location of hard data are shown by gray circles).

![Figure 15. Application of the iCCSIM to a continuous TI shown in Figure 15a. Three realizations conditioned to well data in Figure 15b are shown in Figure 15c. The ensemble average maps of the iCCSIM for the 1-D raster and random paths are shown in Figures 15d and 15e, respectively. Likewise, the conditional variance maps of the realizations generated by the iCCSIM for the 1-D raster and random paths are shown in Figures 15f and 15g, accordingly (the location of hard data are shown by gray circles).]

Table 2. Same as Table 1, But for the 2-D Continuous Field of Figure 14

| Space of uncertainty (between) | iCCSIM: CCSIM | 1:0.99 |
| Pattern reproduction (within)  | iCCSIM: CCSIM | 1:1.25 |
| Ratio (between/within)         | iCCSIM: CCSIM | 1:0.79 |

The MDS-ANODI method is applied in this example. The results are in agreement with the variance maps shown in...
Figures 15f and 15g, where both show that the CCSIM algorithm produces more similar realizations, while the iCCSIM algorithm preserves the similarity and increases the variability between the realizations. The spatial similarities and uncertainty spaces are also numerically expressed in Table 2.

The MDS-ANODI plot is presented in Figure 16. The better pattern reproduction of the iCCSIM algorithm is well illustrated in this plot, as most of the green dots, corresponding to the iCCSIM realizations, are distributed close to the TI (red dot).

5.4. Three-Dimensional Complex Multifacies Modeling

As mentioned earlier, process-based methods mimic the physical process that occur in rock and can produce more realistic images of a large-scale porous medium. In these models, various parameters of the depositional system, including erosion, flow direction, slope, avulsion, aggradation, etc., may be considered in which the equations that govern the actual physical processes are used. Computations with such techniques are, however, extremely demanding. For this reason, process-mimicking methods represent another very promising approach [Pyrcz et al., 2009; Coulthard and Van De Wiel, 2006; Lopez et al., 2008] that can also produce realistic models that do not necessarily follow the governing physical processes. They suffer, however, from conditioning to dense well data, similar to the process-based techniques. Nevertheless, such methods provide an excellent source of models that can be used as the TI. For example, a fluvial meandering channel [Lopez et al., 2008] is presented in Figure 17a.

The proposed iCCSIM algorithm was tested using the image shown in Figure 17a. First, two unconditional realizations were generated for verifying the pattern reproduction, and are shown in Figure 17c. It should be clear that the iCCSIM algorithm produces high-quality models without any data conditioning. Next, the performance of the algorithm using eight wells shown in Figure 17b is demonstrated. Fifty conditional realizations were generated, two of which are shown in Figure 17d. Ensemble average maps for two of the facies (blue and orange) are also provided in Figures 17e and 17f, respectively. As can be seen, the uncertainty around the wells is decreased smoothly.

The improvements by the iCCSIM were quantified. First, the MPC function was calculated for the realizations generated by both the CCSIM and iCCSIM and, then, the results were compared with that of the TI; see Figure 18. They indicate that the connectivity of the main blue channel in the realizations has been improved, and that they mostly mimic the same probability in the TI.

Furthermore, the uncertainty space and the pattern reproduction were quantified using the ANDOI; see Table 3.

The ANODI results indicate that the uncertainty space is shrunk trivially, whereas the pattern reproduction is improved tremendously. These results are also depicted in Figure 18c using the MDS-ANODI plot. The closeness of the uncertainty cloud distributions of the iCCSIM and CCSIM is evident. At the same time, the iCCSIM realizations represent better pattern reproduction, as they are mostly distributed close to the TI.

The effective permeability of the iCCSIM-generated realizations was also compared to that of the TI. To this end, the permeability of the blue, orange, and shale facies were assumed to be 850, 1000, and 10 md, respectively. The effective permeability was calculated in the direction of channels’ distribution. The calculated effective permeability for the TI is 368.84 md while the calculated effective permeability, averaged over 50 realizations, was computed to be 378.36 md.

5.5. Three-Dimensional Object-Based Model

Object-based (Boolean) methods are very useful techniques for providing realistic TIs [Deutsch and Tran, 2002; Pyrcz et al., 2009]. Using some rough geometrical information, such as width, depth, sinuosity, etc.,
one may define a model of a large-scale porous medium. Similar to the process-based methods, object-based techniques do not generate accurate results when conditioned to a large numbers of wells. But, the output of such techniques can be used as the input for higher-order geostatistical methods. An example of
a TI generated by an object-based method is shown in Figure 19a, which represents a highly nonstationary channelized system in which the upper, middle, and lower parts have very different morphological parameters, such as width, depth, levee, and amplitude; see Tahmasebi and Sahimi [2015b]. More cross-sections of the same system are provided in Figure 19b. The model was generated using the 3-D probability maps, extracted from the seismic data shown in Figure 19c. Then, 50 conditional realizations using the available probability maps and the given TI were generated, three of which are shown in Figure 19e. It should be noted that the auxiliary data for describing the TI in Figure 19d were generated using the reverse iCCSIM and the seismic data, in which the input and output are interchanged. More precisely, if the original TI is categorical or discrete but the constraint is continuous, in order to produce a continuous TI (note that the TI in Figure 19a is categorical), we use the seismic data as the TI, and the categorical TI of Figure 19a as the constraint. This generates the various auxiliary data shown in Figure 19c, each of which corresponds to a specific facies in the original TI. The reason that we do this is as follows. Since the available primary inputs (i.e., the TI and the seismic data) are not of the same type, for better representation of the categorical attributes of the TI and making a bridge between it and the seismic data as a continuous variable, their corresponding auxiliary variables for each facies in the TI were generated; see Figure 19c. By using these data sets—the TI, the seismic data, and the auxiliary data—one utilizes the multivariable form of the iCCSIM [Tahmasebi and Sahimi, 2015a]. The produced realizations follow the same distribution of the facies, which was introduced using the seismic data of Figure 19c. For example, the blue facies is mostly located at the top, while the green and red facies are placed in the middle and at the bottom of the realizations, respectively. The quality of the realizations can be checked further by the cross sections shown in Figure 19d. The excellent connectivity of the channels within the conditional simulation indicates the ability of the new method for modeling of complex large-scale porous media. As pointed out in the Introduction, by increasing the number of variables,
Figure 19. Application of the iCCSIM to a complex TI shown in Figures 19a and 19b, obtained from object-based modeling. The extracted probability cubes are shown in Figure 19c. The auxiliary variables were generated using the reverse iCCSIM, shown in Figure 19d. Three conditional realizations are shown in Figure 17e. For exploring the quality, three cross sections from the interior part of the first realization are also provided in Figure 19f.
one must expect more patchiness and discontinuities. The iCCSIM algorithm has, however, the advantage of saving the quality of the realization even when it is conditioned to five variables.

The results in Figure 19 represent very clear improvement of the realizations generated by the iCCSIM over the CCSIM. Such improvements are further quantified by computing the MPC function. The results for the realizations generated by the CCSIM and iCCSIM are compared to that of the TI, and presented in Figure 20. The MPC function was calculated only for the upper blue channels, since their reproduction due to sinuosity and small overlaps are more difficult than the other facies. The overall connectivity of the channels is improved, producing long-range connectivity similar to that of the TI.

The ANODI calculations were also carried out to quantify the uncertainty space and pattern reproduction; see Table 4. Clearly, the uncertainty space has not changed, while pattern reproduction has improved. The MDS-ANODI plot is presented in Figure 20c.

In a fashion similar to the other examples, the effective permeability was also calculated. Thus, the permeability of the main facies in the 3-D object-based model were assumed to be 800, 400, and 200 md for the upper, middle, and lower channels, respectively. The permeability of shale was assumed to be 0.8 md. The effective permeability was calculated in the direction of channels’ distribution. The calculated effective permeability of the TI is 174.93 md, while the calculated effective permeability, averaged over 50 realizations, is 171.65 md.

### Table 4. Same as Table 1, But for 3-D Object-Based System of Figure 19

<table>
<thead>
<tr>
<th></th>
<th>iCCSIM: CCSIM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Space of uncertainty (between)</td>
<td>12.98</td>
<td></td>
</tr>
<tr>
<td>Pattern reproduction (within)</td>
<td>1:1.14</td>
<td></td>
</tr>
<tr>
<td>Ratio (between/within)</td>
<td>1:0.85</td>
<td></td>
</tr>
</tbody>
</table>

5.6. Three-Dimensional Continuous Porosity Field

The example shown in Figure 21a represents a complex continuous porosity distribution. Due to dealing with a continuous variable, providing a comprehensive continuous TI is not
straightforward. Therefore, one should expect a considerable amount of patchiness, even in the unconditional simulation. In this example, eight wells are considered as the hard data that are shown in Figure 21b. Fifty realizations using the CCSIM iCCSIM are generated. For the sake of comparison, a realization was generated by both the CCSIM and the iCCSIM algorithms, using the same seed number in

Figure 21. Application of the iCCSIM to a continuous TI in Figure 21a using eight well data in Figure 21b. Two realizations using the CCSIM iCCSIM are shown, respectively, in Figures 21c and 21d. The ensemble average shown in Figures 21e and 21f and the conditional variance maps shown in Figures 21d and 21g for the CCSIM and iCCSIM are provided in Figures 21e and 21f and 21g, respectively.
the random number generator in order to ensure that the starting points in both algorithms are the same. The results are presented in Figures 21c and 21d. The iCCSIM algorithm produces a high-quality realization, whereas a large amount of patchiness and a number of discontinuities exist in the CCSIM realization. The ensemble average and conditional variance maps for the CCSIM and the iCCSIM algorithm are presented as well. Once again, the iCCSIM provides high-quality realizations and produces symmetric continuity around the well data in Figure 21f; see Figure 21g that shows the symmetry around the wells.

The similarity was also quantified using the ANODI. The results, shown in Table 5, do not indicate any changes in the uncertainty space, while pattern reproduction has considerably been improved. The MDS-ANODI plot representing the spatial uncertainty and pattern reproduction is also depicted in Figure 22. The map shows drastic differences between the pattern reproductions of the iCCSIM and CCSIM. One notes, for example, that in Figure 22 the uncertainty space of the realizations generated by the CCSIM is larger than that of the iCCSIM, which can be due to the fact that the pattern reproduction is not very good and, thus, the realizations represent more variability.

Overall, the uncertainty space of the new proposed iCCSIM algorithm does not manifest a drastic change over the CCSIM algorithm of Part I. Instead, in some cases in which the TI is very complex, the uncertainty space might be trivially changed. More importantly, the new iCCSIM algorithm results in a better reproduction of the heterogeneity patterns.

5.7. Computational Efficiency

Clearly, the computational time of the iCCSIM is higher than that of the improved CCSIM developed in Part I, as it is implemented by an iterative correction loop. As emphasized throughout the paper, however, the error map reduces the computational burden. Thus, overall, based on the number of iterations and the existing errors in a typical realization, the computational time is increased only slightly over the method in Part I. For example, the computation time for generating a single realization for the TI shown in Figure 21 is only 4 CPU minutes if two iterations are used. As another example, the computation time for the realizations shown in Figure 19 is increased to 7 min with the iCCSIM from the CPU time of the CCSIM of Part I that was 3 CPU minutes. These should be compared with the computational times two other MPS algorithms. Producing one realization for the TI in Figure 19 would take 210 and 450 CPU minutes by the SNESIM and FILTERIM algorithms, respectively.

6. Conclusions

Use of the pattern-based techniques for modeling of large-scale porous media is increasing. This is due to the fact that, compared to the pixel-based approaches, the complex spatial features are reproduced more accurately by such methods in a very short computational time. By increasing the complexity of the TI and the number of the conditional data sets, including the point and secondary data, however, the quality of the realizations decreases and numerous discontinuities and patchiness may emerge. One also needs to have a method that can extract the maximum amount of information from a TI, which can help identifying the matched pattern and, consequently, decrease the discontinuity.

In this paper, a new iterative algorithm based on the CCSIM method and graph-cut technique—called

| Table 5. Same as Table 1, But for 3-D Object-Based System of Figure 21 |
|--------------------------|----------------|----------------|
| Space of uncertainty (between) | iCCSIM: CCSIM | 1:0.98 |
| Pattern reproduction (within)  | iCCSIM: CCSIM | 1:1.32 |
| Ratio (between/within)          | iCCSIM: CCSIM | 1:0.74 |

Figure 22. Same as Figure 11, but for 3-D continuous field of Figure 21.
Acknowledgments
This work was supported in part by the RPSEA Consortium. The authors would also like to thank the constructive comments from three anonymous reviewers and the associate editor who helped us improve the original manuscript. This work is based on the CCSIM code available at: https://github.com/pejmant/MS_CCSIM. All data and digital content in this manuscript can be accessed by sending an e-mail to tahmasebi.pejman@gmail.com

References
Gonzalez, R. C. (2009), Digital image processing, Pearson Education India.

iCCSIM—was introduced. The core idea of iCCSIM lies in using an error map to guide the algorithm to repair the patchiness and mismatches caused by the hard data. The error map can be provided by the CCSIM algorithm. The use of the iCCSIM and a distance function richer than the Euclidean distance, such as the distance transform and the gradient-based methods improve the pattern reproduction significantly.

To verify the performance of the new method, a series of models that have been generated by using process/object-based methods were selected as the training image (TI). The iCCSIM was demonstrated to be able to successfully produce high-quality conditional realizations for multifacies and continuous TIs in both single and multivariable simulations.