



Investigating extreme scenarios with multiple-point geostatistics and variance maximization

Mohammad Javad Abdollahifard¹ · Grégoire Mariéthoz² · Hesam Soltan Mohammadi^{3,4}

© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

In many geoscience applications, the data extracted from environmental variables are very limited. Multiple-point geostatistical (MPS) approaches simulate these variables and associated uncertainties at unknown locations by using an exemplar model for the field, called the training image (TI). Existing MPS approaches aim at simulating the field in a way consistent with both available conditional data and TI properties. The inevitably limited size of the training database usually leads to an underestimated variability between different realizations as compared to the variability of the real phenomenon. Furthermore, in over-conditioned regions, patch-based methods often tend to paste the same patch in all realizations. In this paper, we suggest an optimization-based approach for MPS simulation that simulates a bunch of realizations simultaneously. In addition to maintaining consistency with both conditional data and TI properties, the proposed method aims at maximizing the variability between different realizations. Our experiments show that the proposed strategy enhances the variability of the realizations to better conform with real variabilities. The idea of targeting variance maximization can potentially be applied to other MPS simulation methods by simulating a bunch of realizations simultaneously with a constraint to avoid similar patterns at the same location in different realizations.

Keywords Environmental modeling · Variability · Verbatim copy · Optimization-based simulation

1 Introduction

Multiple-point geostatistics are used in an increasing number of applications to quantify spatial uncertainty (Mariethoz and Caers 2014). It is especially useful when one is interested in structured features such as connected patterns. Such patterns are typically described by training images. Many algorithms have been proposed for MPS simulation over the last two decades (Hu and Chugunova 2008). Some of them store catalogues of patterns extracted

from the training image, which are used to generate simulations honoring the same patterns frequencies (Abdollahifard and Faez 2013; Strebelle 2002). Other approaches do not create catalogues of patterns, but instead look up for matching patterns directly in the training image itself. This is achieved by scanning the TI in a random order (Abdollahifard and Faez 2014; Mariethoz et al. 2010), or by using fast convolutions.

Besides the way they store patterns, MPS algorithms also differ in the ways they proceed to generate stochastic realizations. Certain methods are known as pixel-based algorithms because they determine the value of a single simulation grid cell at a time. Certain approaches simulate groups of cells at the same time (Rezaee et al. 2013), and are therefore known as patch-based (Arpat and Caers 2007; Mahmud et al. 2014; Sharifzadeh et al. 2018; Zhang et al. 2006). Further, some methods use sequential simulation (Abdollahifard 2016), some use iterative simulation (Li et al. 2016; Tahmasebi and Sahimi 2016), and others use simulated annealing (Peredo and Ortiz 2011).

✉ Mohammad Javad Abdollahifard
mj.abdollahi@tafreshu.ac.ir

¹ Electrical Engineering Department, Tafresh University, Tafresh 39518 79611, Iran

² Institute of Earth Surface Dynamics, Université de Lausanne, Lausanne, Switzerland

³ School of Mining Engineering, College of Engineering, University of Tehran, Tehran, Iran

⁴ Mine Environment and Hydrogeology Research Laboratory (MEHR Lab.), University of Tehran, Tehran, Iran

While these algorithms differ in many ways, all of them essentially aim at recomposing patterns found in the TI such as to create stochastic simulations that honor two properties:

1. The spatial structure of the training image must be respected (this can be defined in a variety of ways ranging from experimental variograms to multiple-point histograms)
2. A given set of conditioning data has to be honored.

In this paper, we consider the introduction of a third property, namely that the simulations should span an uncertainty space that is as wide as possible.

Indeed, the main limitation of all geostatistical approaches based on training images is that the variability that can be obtained is limited by the size and diversity of the training images used. This has been shown on applications of MPS to the simulation of very long hydrological time-series, where there is a tendency not to reproduce extreme events that do not occur frequently enough in the training image (Oriani et al. 2014). In general, it is found that a trade-off exists between pattern reproduction and variability. Essentially, there is an intrinsic limit to the variability of patterns that can be produced from a finite training image. It has been shown that simulations based on training images cannot be assimilated to random fields, except if the training image is impractically large (Emery and Lantuéjoul 2014). One corollary of this variability limitation is the phenomenon of verbatim copy that occurs when simulations are based on a small training image with complex patterns: the corresponding complex structures produced in the simulation are often copied identically as in the training images.

TIs may either be obtained from a densely sampled field having similar behavior to the target field (Rasera et al. 2019), or conceptually synthesized based on elementary shapes and parameters proposed by geology experts (Maharaja 2008). In the latter case, we can easily produce large training images using software tools. Larger TIs are expected to show higher variability. However, since these TIs are synthesized based on the same concepts, they will introduce minimal innovations. Furthermore, increasing the size of the TI leads to an increase in the computational burden. In the case of extracting the TI from a densely sampled field, increasing the TI size usually involves spending much more time and money. This is particularly much more demanding when in situ data acquisition tools are used, as compared to remote sensing methods (Abdollahifard et al. 2016). In both cases, the TI aims at describing the behavior of a natural texture. Natural textures, however, are usually much more complex and diverse than being describable in a limited TI. For example,

even if one collects dozens of images of wood textures, it is always possible to identify new patterns in a new image.

One proposed solution has been to design ways to generate new patterns that do not exist in the training image (Honarkhah and Caers 2010), but this comes at the risk of creating non-realistic structures and does not explicitly address the issue of too low variability between realizations. Most multiple-point algorithms have been designed to enhance patterns reproduction, but doing so generally entails some to extent that the realizations look similar. To the best of our knowledge, no report has been presented in which the increase in the variability of realizations has been directly targeted. Only Rezaee et al. (2015) attempted to create and use a larger training image with unconditional simulation from the original training Image. However, as the new image derives from the original image, this method is not optimally suited to generate entirely new patterns that are not present in the training image. In addition, enlarging the TI can increase the computational cost dramatically which is one the main bottlenecks in the multiple point simulations (Abdollahifard and Nasiri 2017; Tahmasebi 2018).

In this research, we attempted to explicitly include the variability enhancement of realizations in the problem formulation as a third property and this modification will occur in an optimization-based framework. Optimization-based methods fill in the simulation grid randomly and then iteratively improve the grid according to an objective function. In contrast to the existing MPS algorithms which generate each realization independently, in this study all realizations are synthesized simultaneously. In addition to the fulfillment of the goals as mentioned earlier, the objective function is considered to increase the variability of different realizations. In this way, the realizations generated by the proposed method will be more diverse than previous methods, which is closer to reality and can compensate for some of the bias caused by the limited training image.

2 Background on optimization-based methods

While optimization-based methods were initially developed for texture synthesis applications (Kopf et al. 2007; Kwatra et al. 2005), recently they are generalized to become applicable to the conditional simulation problems (Abdollahifard and Ahmadi 2016; Kalantari and Abdollahifard 2016; Pourfard et al. 2017; Yang et al. 2016). Unlike sequential methods, optimization-based methods do not have a clear sequence in scanning simulation grids, but all parts of the simulation grid are scanned simultaneously. These methods are iterative. First, the blank parts of image

filled in randomly. Then in each iteration, the algorithm tries to modify the simulation grid in such way honoring the hard conditioning data while imposing the TI behavior in the grid.

The data-event extracted at each grid point contains values from the previous iteration and then all data-events can be processed in parallel. The optimization is carried out through a two-step expectation–maximization (EM) approach. In the Maximization phase, for each data-event, the whole TI is searched to find the best match. Then, in the Expectation phase the found matches are replaced in a new grid by an averaging operator. To prevent edge effects and preserve consistency, remarkable overlap is considered between adjacent data-events. In the Expectation phase, the average of the overlapping values of the found matches will be replaced in the grid. Then, the new grid is replaced with the old one and the process is repeated to a maximum number of iterations.

Such methods usually work in a multi-scale framework. At first, they start by simulating a coarse-resolution image based on a coarse counterpart of the TI. Then, they proceed by interpolating the synthesized image into a finer resolution to serve as initialization for a similar iterative refinement process based on a finer TI.

In the Maximization (match finding) phase, usually the whole TI is scanned to find the best match. This is done by computing a distance between the data-event and the TI patterns and then, finding the pattern with minimum distance. For conditional simulation, the importance of hard data is emphasized through computing a weighted distance and assigning higher weights to them as compared with synthetic values. Searching the whole TI for each data-event makes the optimization-based methods very computationally demanding. To speed up these methods different proposals were made in the literature including random search techniques like PatchMatch (Yang et al. 2016), GPU-based acceleration (Pourfard et al. 2017), and dictionary decimation (Kalantari and Abdollahifard 2016). For categorical variables, it is possible to form search trees from the TI and then search in the tree more effectively (Strebelle 2002).

To model the uncertainties associated with conditional simulation, patch-based methods usually proceed by finding a certain number of best matches for each data-event, picking one of them randomly and placing the selected one in the simulation grid. In optimization-based methods, however, the variability of realizations is driven by different random initializations.

In the coarsest scale, the TI and SG are usually very small and comparable with the data-event in size. Then, one data-event covers a considerable portion of the SG and often contains considerable number of conditioning data. Since the conditioning data are usually weighted by very

strong coefficients, the random initializations lose their effect in this situation. Therefore, the matching phase often leads to the same results (patches) and the methods tend to produce very similar realizations which are a verbatim copy of the TI.

To alleviate verbatim copy problem Yang et al. (2016) suggested employing an enlarged TI resulting in a significant increase in computations. Kalantari and Abdollahifard (2016) suggested increasing the diversity of training patterns by multiplying TI patches with an arbitrary quantity or even allowing a weighted combination of two (or multiple) patches to produce a new patch. To prevent producing patterns inconsistent with the TI, they have proposed some tricks including local histogram matching for stationary images.

3 Basic approach

In this paper, we have adopted the method of (Kalantari and Abdollahifard 2016) as the base of our work. In this section, the basic algorithm is presented in some details. Without loss of generality, let us assume that both the field of interest and the TI are two-dimensional images and we use square-shaped patches. The method, however, can easily be generalized to 3D fields. Let us denote $\sqrt{n} \times \sqrt{n}$ training patterns extracted from the training image I with $\tilde{\phi}_x$, where $x \in \Omega_I$ is a two-dimensional position vector designating the centroid of the patch and Ω_I denotes the support of I . The vectorized form of these patches are denoted by $\phi_i \in \mathbf{R}^n$. Likewise, consider that a patch (data-event) \tilde{p}_y of the same size is extracted around point $y = [y_1, y_2]^T$ in the simulation grid J ($y \in \Omega_J$). Note that, in optimization-based methods these patches do not contain any empty region, and even at the beginning, the SG is filled with some random values. In this patch, also there may be a number of hard conditioning data which are highly important and must be respected during refinement. The algorithm seeks for a training patch which minimizes the following weighted distance:

$$E(x) = \sum_{j=1}^n w_j (\mathbf{p}_y(j) - z_x \phi_x(j))^2, \quad (1)$$

where $\mathbf{p}_y(j)$ and $\phi_x(j)$ denote the j th element (pixel) of the patches \mathbf{p}_y and ϕ_x and w_j denotes the weights for different pixels. For conditioning data points the weight is set to a large value (e.g. 100) and for other points the weight is considered as 1. z_x is an optimum scalar multiplier which minimizes E . By taking the derivative of E with respect to z and setting it to zero, z_x will be obtained as follows:

$$z_x = \operatorname{argmin}_z \{E(x)\} = \frac{\sum_{j=1}^n w_j \phi_x(j) p_y(j)}{\sum_{j=1}^n w_j \phi_x^2(j)}. \quad (2)$$

In fact, instead of simply searching in the training patterns to find a match, at first the method attempts to make each training patch as consistent as possible with the query patch by multiplying an optimum quantity. For example, assume that we are searching for a patch that is composed of two halves with two different values. The right half is filled with zeros and the left one is filled with 80 s. A patch with the same structure but with zeros in the left side and 10 s in the right side is not a good match for the query patch. This algorithm allows modifying the second patch by multiplying it with an optimum quantity (8 in this example) and then this will be a perfect match. This strategy will increase our chance for finding a good match particularly for rare patterns.

After computing optimum z , we find the best match as $p_y^* = z_x^* \phi_{x^*}$, where

$$x^* = \operatorname{argmin}_x \{E(x)\}. \quad (3)$$

This can be continued (for maximum of s times) by computing the residue, $r = p - p^*$, and then finding another match for r (s is called the sparsity factor). However, in this paper we set $s = 1$, not allowing excessive pattern variability by combining patches.

After finding a match for each patch of the SG in the maximization phase, the matches are replaced in corresponding locations in a new grid J' . In order to keep consistency between adjacent patches, they are considered with a large overlap. In fact, the spatial shifts between subsequent patches are as small as one pixel. Then, for each point of the new grid, there may be tens of candidates in the found matches. This is usually handled by filling each point with the average of overlapping values in the expectation phase. After completing the new grid, the previous grid is replaced with the new one at once ($J \leftarrow J'$) and the process continues until a maximum number of iterations is reached.

Although allowing the weight z_x increases the diversity of training patterns, it may result in realizations which their brightness is not consistent with the TI in some regions. To prevent such deviation, after each iteration the histogram of the realization is forced to match that of the corresponding TI (Peyré 2009). For stationary images, local histogram matching leads to more consistent realizations as shown by (Kalantari and Abdollahifard 2016). After completing the simulation grid in one scale, the image is interpolated to the next scale and used as an initialization for a similar process in that scale.

Due to the averaging of overlapping values, optimization-based methods usually suffer from a slight blurring

effect. The averaging operator generates new values that may be different from all operands. In contrast, the median operator selects one of the operands as the output and hence slightly reduces the blur effect. For computing the median of a sequence of numbers, one typically needs to keep all of them in memory. To reduce memory Lyon and Ward (1980) suggested a sequential median calculation method which guarantees to find the exact median by using half of the memory required for storing all values, and produces acceptable outputs even using less memory. Nevertheless, median calculation is still much more memory-demanding than averaging. It should be noted that, the method (Lyon and Ward 1980) is prone to fail if the consecutive values are highly correlated. A good way for preventing this problem is to randomize the order of scanning the image. Note that, in optimization-based methods the realizations are completely independent of the order of the scan.

4 The proposed method

In this paper, we aim at generating a number of realizations, J^k s ($k = 1, \dots, K$), simultaneously by considering their relationship and trying to maximize local variability among them. The realizations are obtained by minimizing the following energy function:

$$\varepsilon(J^1, \dots, J^K) = \sum_{k=1}^K \{e_i(J^k) - \lambda e_v(J^k)\}. \quad (4)$$

$e_i(J^k)$ evaluates the inconsistency between J^k and the TI and $e_v(J^k)$ evaluates the cumulative local variability caused by the J^k . In fact, our goal is to minimize the inconsistency between realizations and the TI, and maximize the variability between different realizations. λ is a positive quantity between 0 and 1 which controls the relative importance of the two terms. The inconsistency term is computed by comparing each data-event in J^k with its best match in the TI:

$$e_i(J^k) = \sum_y d(p_y^k, \phi_x^*), \quad (5)$$

where $p_y^k \in R^n$ is the patch extracted around location y in J^k and $\phi_x^* \in R^n$ is its best match found in the TI and $d(., .)$ is a distance function.

The variability term is formed by comparing each patch in each realization with the average of all co-located patches in all simultaneously evolving realizations:

$$e_v(J^k) = \sum_y d(p_y^k, \bar{p}_y), \quad (6)$$

or equivalently:

$$\varepsilon_v(J^k) = \sum_y d(\phi_x^*, \mathbf{p}_y), \quad (7)$$

where:

$$\bar{\mathbf{p}}_y = \frac{1}{K} (\mathbf{p}_y^1 + \dots + \mathbf{p}_y^K). \quad (8)$$

The objective function of Eq. (4) can be rewritten for the patch level as follows:

$$E(x) = \sum_{j=1}^n w_j (\mathbf{p}_y^k(j) - z_x \phi_x(j))^2 - \lambda \sum_{j=1}^n (z_x \phi_x(j) - \bar{\mathbf{p}}_y(j))^2. \quad (9)$$

In the above equation, the importance of satisfying hard conditioning data is emphasized by assigning higher weights (w_j) to them. While the first term attempts to enforce conditioning data and encourage patterns similar to those of the previous iteration, the second term attempts to maximize the distance of the match with the average of the patches in the same location. λ controls the relative importance of both terms. Higher λ values encourage more diversity but can also result in instability.

For minimizing the objective function of Eq. (4) an iterative EM-like approach is employed. At first all realizations are filled randomly at unknown locations. Then, assuming fixed realizations J^k , $k = 1, \dots, K$ (and equivalently fixed patches \mathbf{p}_y^k for all k and y), we attempt to find the optimum values for z and ϕ which minimize Eq. (9). Afterwards, the patches are updated by $\mathbf{p} = z\phi$. The whole realizations are refined by computing the average of the overlapping values of different patches. This process continues to a number of iterations. The detailed pseudo-code for the proposed algorithm is given in Table 1.

The best ϕ should be found by searching in a pattern data-base extracted from the TI. The optimum value for z can be found analytically by setting the derivative of Eq. (9) w.r.t z to zero:

$$z_x = \frac{\text{num}_x}{\text{den}_x} = \frac{\sum_{j=1}^n (w_j \mathbf{p}_y^k(j) - \lambda \bar{\mathbf{p}}_y(j)) \phi_x(j)}{\sum_{j=1}^n (w_j - \lambda) \phi_x^2(j)}. \quad (10)$$

For each patch \mathbf{p}_y^k , we need to compute z_x for all $x \in \Omega_I$, and then put it in Eq. (9) and finally find the patch x for which the energy $E(x)$ is minimized. Putting z_x from Eqs. (10) in (9) and simplifying yields in the following direct form for E :

$$E(x) = \frac{-\text{num}_x^2}{\text{den}_x} + C \\ = \frac{-\left[\sum_{j=1}^n (w_j \mathbf{p}_y^k(j) - \lambda \bar{\mathbf{p}}_y(j)) \phi_x(j) \right]^2}{\sum_{j=1}^n (w_j - \lambda) \phi_x^2(j)} + C. \quad (11)$$

Note that, in this equation C does not depend on x and thus can be ignored when searching for optimum x . Using the latter formulation, we can avoid unnecessary computations by directly computing the energy for all x , finding the x^* which minimizes $E(x)$, computing z_{x^*} and finally computing $\mathbf{p}^* = z_{x^*} \phi_{x^*}$ and putting it in the corresponding location around y in k th realization.

To see the effect of decimated and learned dictionaries, interested readers are referred to (Kalantari and Abdollahifard 2016). The dictionary (data-base) of training patterns employed in this paper is a full exemplar dictionary composed of all $\sqrt{n} \times \sqrt{n}$ patches of the TI. In practice, in this case it is not required to form such a big dictionary in memory. Instead, one can directly work with the TI by putting the coefficients in a square window and then filtering the TI (for the numerator) or squared TI (for denominator) with the window.

For conditional simulation, the conditioning data should be rescaled to different resolutions. In a coarse scale, each pixel is equivalent to a number of pixels contained in a square in fine resolution. All hard conditioning data available in such squares are averaged and put in the corresponding pixels in coarse scale. Since these squares are typically incomplete, the averages computed are not precise (they have some uncertainty) and for coarser scales the uncertainty is higher. For this reason, we have changed the weights of conditioning data from small to large values, for coarse to fine scales. As mentioned previously, in coarse scales, each patch covers a significant portion of the SG (and conditions) and pushing the conditioning data too hard may lead to finding the same match in all realizations, reducing the variability significantly. By employing the above-mentioned weight changing strategy this problem is also alleviated to a great extent.

Table 1 shows the pseudo-code of the proposed algorithm. It should be noted that at each iteration all realizations are updated simultaneously. Consequently, all patches of each realization and all realizations can be processed in parallel. This could be employed for speeding up the algorithm on a graphical processing unit (GPU).

5 Evaluation of the proposed method

5.1 Multi-Gaussian case

Our first test is performed on a random field having a particular mathematical model. This allows us to compute real variabilities through mathematical simulations and compare them against variabilities obtained from MPS simulations. Here we use multi-Gaussian random fields

Table 1 The pseudo-code of the proposed algorithm

```

■ for  $l = 1, \dots, L$  (for all levels of the pyramid),
    ■ map the conditional data in a grid of appropriate size,
    ■ if  $l = 1$  (coarsest resolution),
        ■ for  $k = 1, \dots, K$  fill  $J^k$  randomly in unknown locations,
    ■ else ,
        ■ resize (by bilinear interpolation) realizations obtained in level  $l - 1$  and use the resized realizations as initializations for  $l$ th level,
    ■ end if.
    ■ for  $i = 1, \dots, N_{iter}$  (for a number of iterations),
        ■ (parallel) for  $k = 1, \dots, K$  (for all realizations),
            ■ (parallel) for  $y \in \Omega_J$ ,
                ■ extract  $\mathbf{p}_y^1, \dots, \mathbf{p}_y^K$  and compute  $\bar{\mathbf{p}}_y$  using equation (8),
                ■ determine  $\mathbf{w} \in R^n$  based on  $\mathbf{p}_y^k$  (assign high weights for hard conditions and low weights for the others),
                ■ search the training data-base to find the patch  $\phi_{x^*}$  which minimizes (11),
                ■ compute  $z_{x^*}$  for  $\phi_{x^*}$ ,
                ■ compute updated  $\mathbf{p}_y^k: \hat{\mathbf{p}}_y^k \leftarrow z_{x^*} \phi_{x^*}$ ,
            ■ end for.
        ■ end for.
        ■ for all  $k$  update  $J^k$  by putting updated patches together and taking average from overlapping values.
    ■ end for.
■ end for.

```

generated using the fast Fourier transform moving average (FFT-MA) (Le Ravalec et al. 2000).

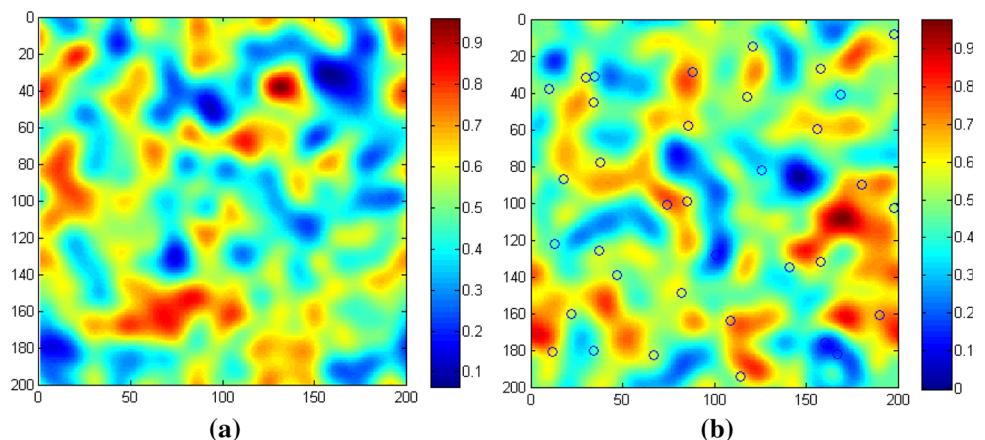
In this study, two unconditional multivariate Gaussian random fields are generated using the same parameter settings and employed as TI (Fig. 1a) and target field (Fig. 1b). The parameters include mean and variance of the simulation and the covariance model that completely defines the behavior of the process by influencing stationarity, isotropy, smoothness, periodicity and correlation length. The first experiment is carried out with a correlation length of 40, both in horizontal and vertical directions. 32 samples points were extracted from the target field in random locations marked with circles.

Different simulation methods were employed for completing the grid, with 100 realizations produced by each method. The first method is based on the true mathematical

model and hence is expected to show the real variability. Unlike this method, all the remaining ones are founded on the MPS principles, i.e. attempt to complete the grid by employing the TI patterns, with the optimization-based approach presented in Sect. 4, for different values of $\lambda \in \{0, 0.2, 0.3, 0.4, 0.5\}$. $\lambda = 0$ represents an ordinary MPS simulation method. Note that in these tests the TI is relatively small, a situation where MPS approaches typically present a significant loss of variability between realizations.

In Fig. 2, for each value of λ , two realizations obtained by the proposed method are presented. It has been previously mentioned that the use of large values for λ can cause the algorithm to become unstable. The realizations presented in Fig. 2d and e show that $\lambda = 0.4$ and $\lambda = 0.5$ in some places cause an undesirable and inconsistent

Fig. 1 Two images generated by a multi-Gaussian model with correlation length of 40 in vertical and horizontal directions



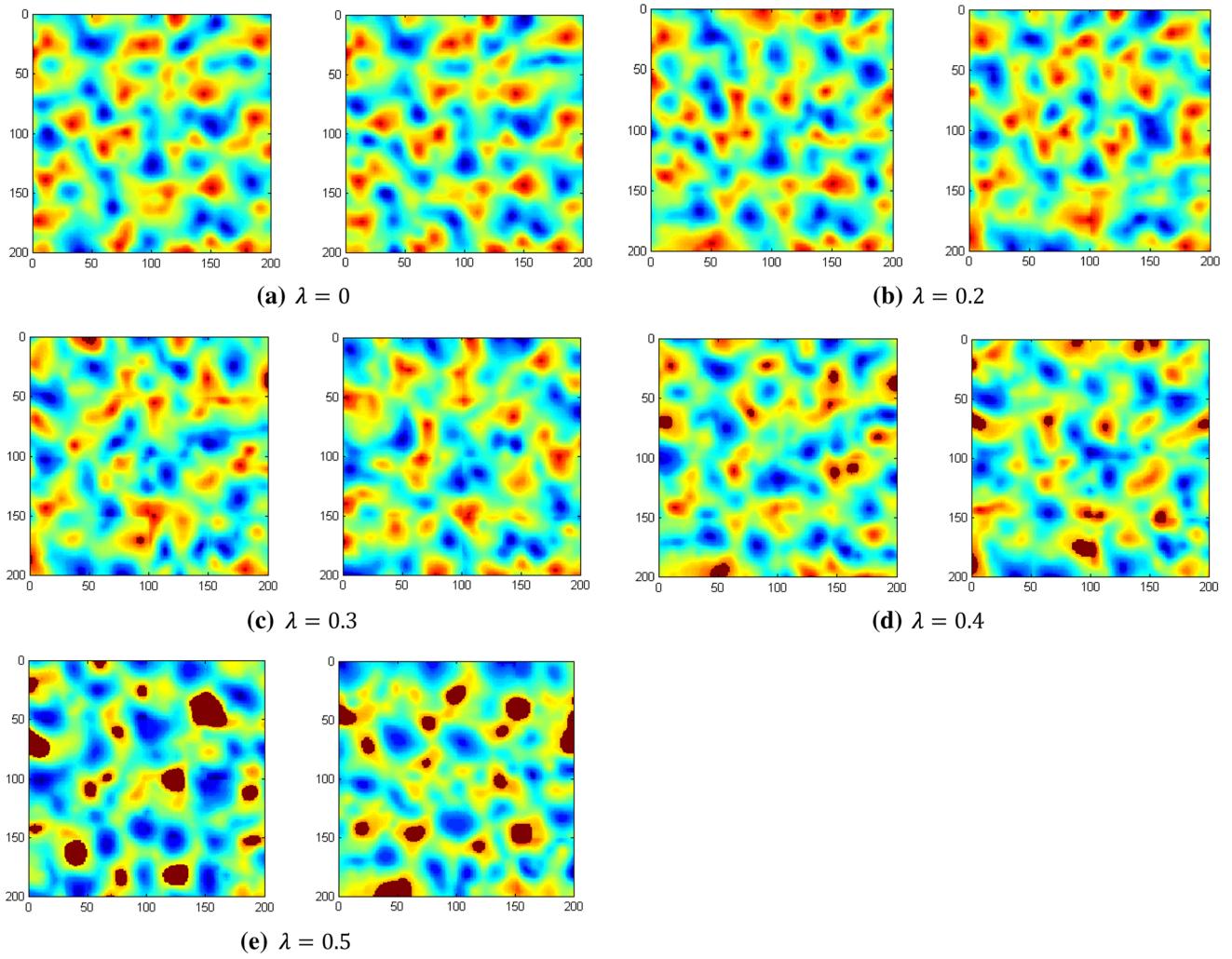


Fig. 2 Two samples of realizations generated by the proposed method for different λ

saturation. This fact is clearly visible in the variograms depicted in Fig. 3 for different realizations. The variograms show that excessive emphasis on variance maximization through increasing λ adversely affects the reproduction of the statistics. One can use this analysis for tuning the value of λ : the best value for λ is the largest value which keeps the reproduced properties in a tolerable range. In Fig. 4 the variance maps of all 100 realizations are illustrated for different values of λ . These maps confirm an increase in local variances by increasing λ .

In order to have a clear physical understanding of this example we can assume that these images show digital elevation maps (DEMs) of hilly areas. Suppose that a hiker wants to reach the right side of the partially known field from its left side with the least possible effort, and our goal is to examine the range and distribution of minimum required effort while considering uncertainties. In geoscience applications, this objective can be replaced with

any (possibly more complex) transfer function, such as for example a flow function used in hydrogeology.

Our goal is to find the shortest path between any starting point on the left side and any arrival point on the right side. The cost of a path is defined as the cumulative uphill elevation changes through the path, and termed as uphill effort. For a fully known grid, the lowest cost (minimum uphill effort) can be efficiently computed through dynamic programming optimization. For the partially known field (a field having 32 points with known elevations), at first, the simulation grid needs to be completed and then the minimum uphill effort is computed for the obtained realization. By repeating this for all of the realizations obtained using either mathematical model or the proposed method, we can find and plot the distribution and range of minimum required efforts.

Let us denote a conditional realization by $J(y), y \in \Omega_J$. In the leftmost column, we start the cost of all pixels with zero. To ensure path continuity, at each step the hiker is

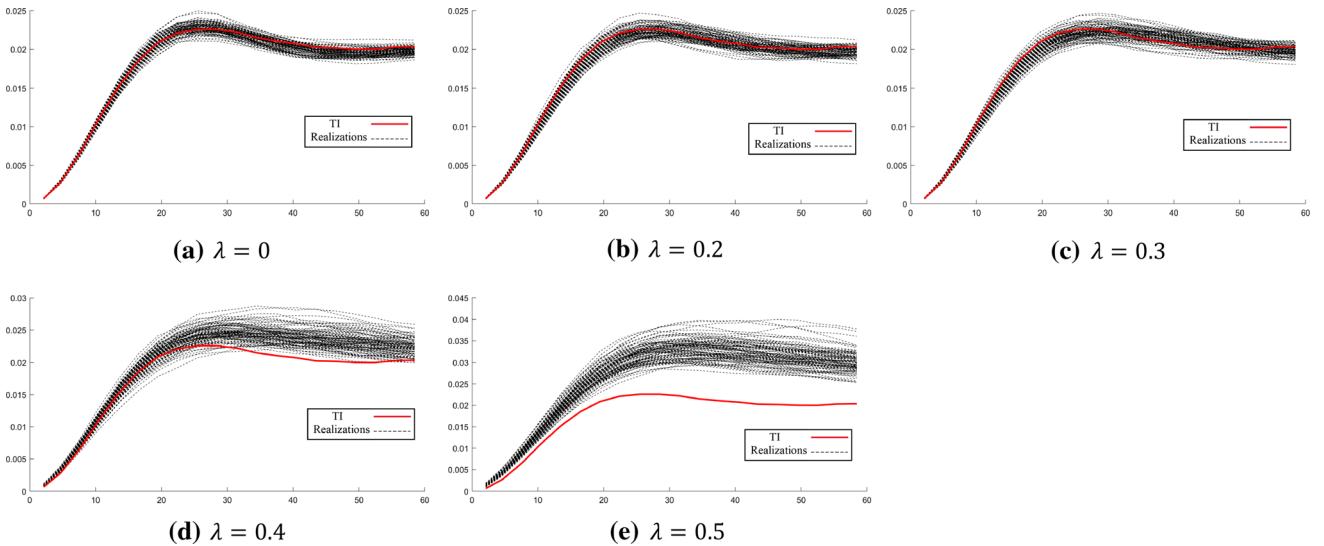


Fig. 3 each pane shows the variograms of 100 realizations obtained using the proposed method based on the TI and conditional data of Fig. 1 for different values of λ in black, along with the variogram of the TI in red. Increasing λ adversely affects the reproduction of the TI properties

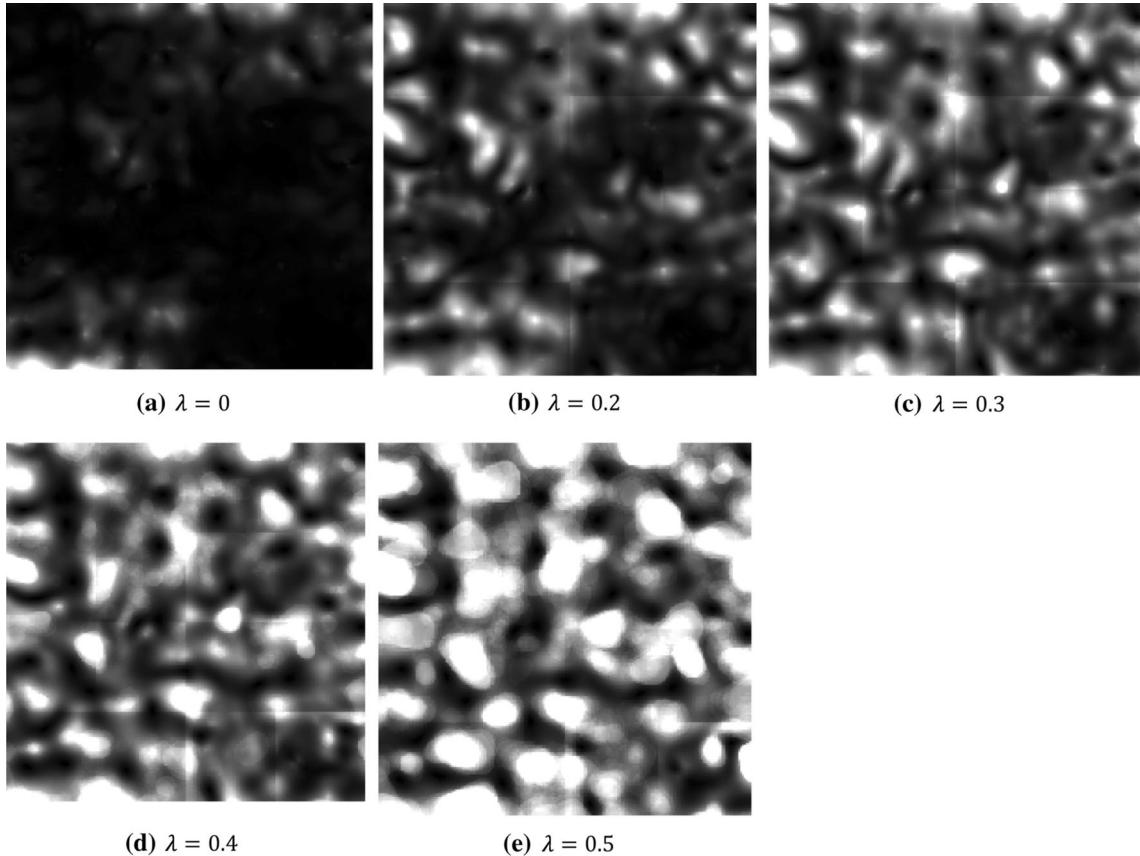


Fig. 4 Variance map of 100 realizations generated with the proposed method. Local variance increases by increasing the λ

only allowed to move to one of the three neighboring pixels of his/her current position (pixel) in the next column. The set of pixels that can be visited in one step from y_i is

denoted by N_{yi} . If $y_j \in N_{yi}$, then the transition cost from y_i to y_j is obtained from the following equation:

$$C_{ij} = \max\{0, J(y_j) - J(y_i)\}$$

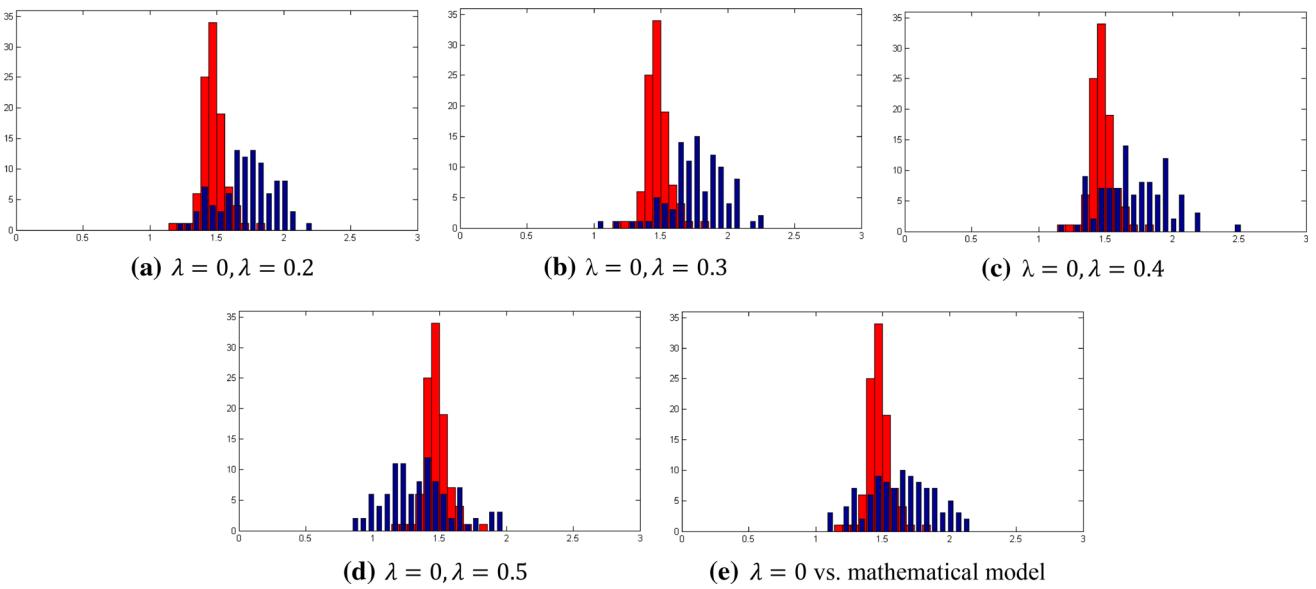


Fig. 5 Histograms of minimum uphill efforts for 100 realizations, red histograms are generated by ordinary MPS ($\lambda = 0$) and blue histograms are generated by different λ values, e shows the histogram of real field generated through mathematical model

In other words, while downhill movements add no costs to the path, the cost of uphill movements is proportional to their associated elevation change. The cost of a path equals the sum of all individual steps.

Instead of examining all possible paths exhaustively to find the best one, dynamic programming proceeds by breaking the problem down into a set of simpler subproblems (i.e. finding optimum sub-paths). For each realization the minimum uphill effort is computed. Then the corresponding histogram of efforts for each method is formed.

Figure 5e compares the effort histograms of 100 realizations obtained from mathematical model (blue) against ordinary MPS ($\lambda = 0$) (red). As expected, the ordinary MPS method shows significantly lower variability compared to the real model. As will be illustrated shortly, the reason lies in the limited size of the TI, which does not cover the full diversity of all possible patterns required to appropriately model the behavior of the field. As shown in Fig. 5a-d, increasing λ leads to an increase in the variabilities of the effort histograms and compensates the limited size of the TI, resulting in a significant improvement while attempting to investigate the extreme scenarios. While the variability for $\lambda = 0.2$ and 0.3 is close to the real mathematical model, higher values for λ leads to realizations whose efforts lie beyond the range of real values.

5.2 Sensitivity on the TI size and anisotropy

As noted before, the main reason that MPS methods cannot appropriately model the variations of real fields is the limited number of training patterns available in the TI.

With a limited size TI, the available patterns are not sufficient for describing all variations of the real phenomenon. In this section, the goal is to investigate how much increasing the training image dimensions can improve the problem. This time, instead of using the 200×200 TI shown in Fig. 1a, a 600×600 training image is used with the same parameters (Fig. 6a).

Histograms of uphill efforts in Fig. 6 show that using an enlarged training image has made it possible to appropriately model the real variations just by using ordinary MPS and without the use of the proposed method. In practical applications, TIs are limited by the amount of data available. In addition, it should be noted that increasing the size of the training image leads to a considerable increase in the computational cost. For the simple and smooth image considered in this experiment, TI enlargement works quite well. However, for more complicated and structured textures, in order to have a training image with all possible patterns included, it would probably be necessary to use a very large TI and MPS simulation would then be computationally unfeasible.

We have repeated the experiment in Sect. 5.1 with anisotropic TI and sampling field (Fig. 7). The TI and the sampling field are constructed similar to the previous example, but with different correlation length values in horizontal and vertical directions. This time the correlation length in the horizontal and vertical direction are 50 and 20, respectively (Fig. 7). Figures 8, 9, 10 and 11 are produced similar to Figs. 2, 3, 4 and 5, and confirm the results of the previous experiment.

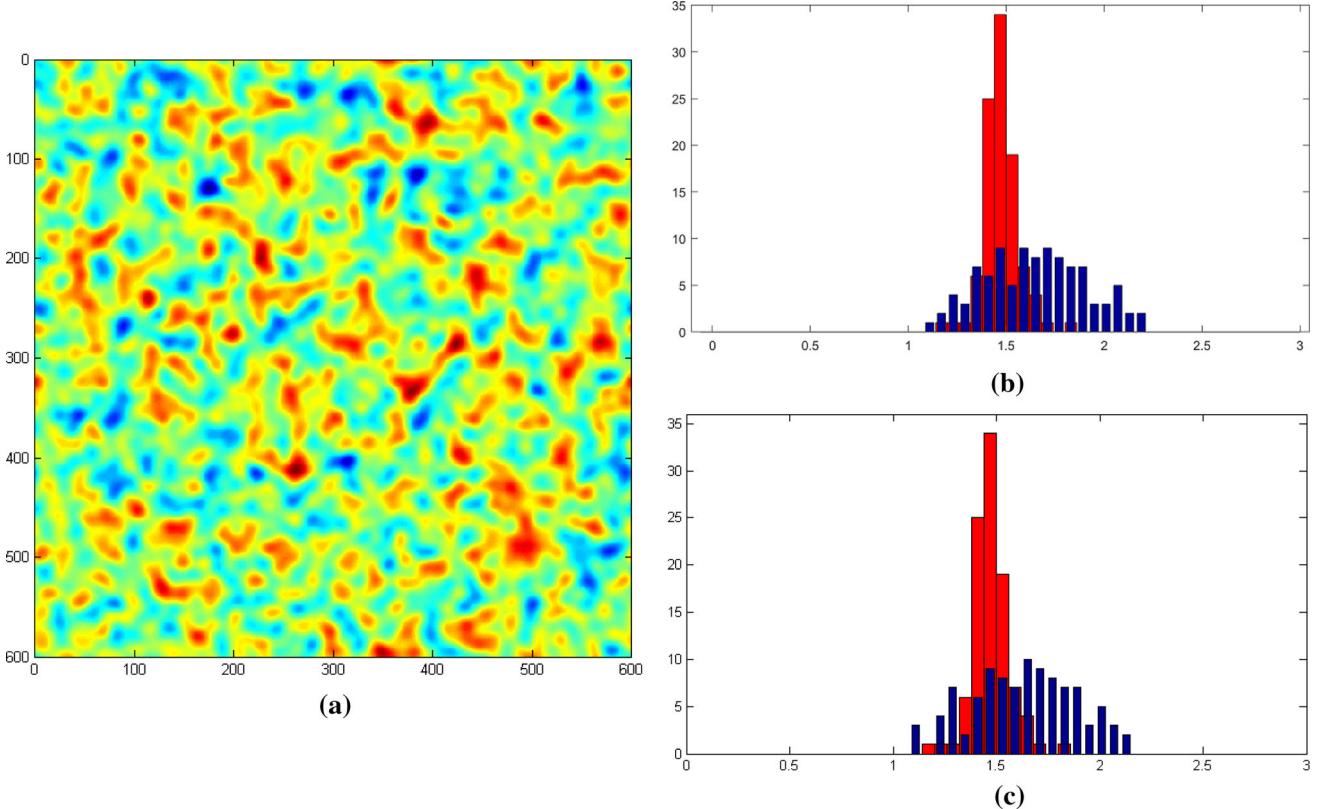


Fig. 6 **a** Large TI (600×600) to compensate for the lack of variance in realizations. **b** Histogram of uphill effort for 100 realizations, red: small TI with $\lambda = 0$, blue: realization generated by mathematical model. **c** Histogram of

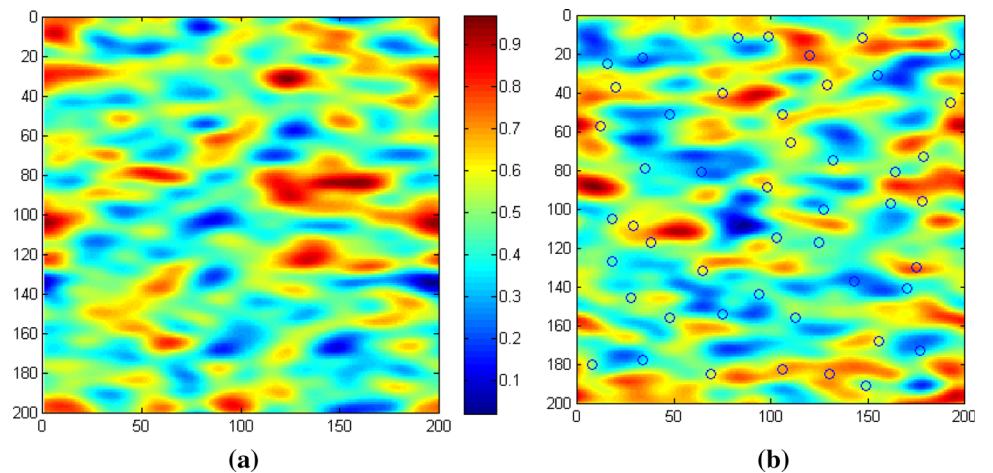
uphill effort for 100 realizations, red: small TI with $\lambda = 0$, blue: realization generated by mathematical model

5.3 How to choose λ

As depicted in previous examples, there is a trade-off between innovation and consistency of the realizations. For low values of λ , realizations are highly consistent with the TI, but show low variability. For high values of λ , the realizations show high innovation but suffer from low consistency with the TI. Recently, a method was developed

for quantitative evaluation of innovation and consistency of MPS realizations (Abdollahifard et al. 2019). For consistency evaluation, it relies on comparing the histograms of particular texture descriptors called local binary patterns (LBP) between the TI and the realization. For evaluating the innovation capability, it partitions the realizations into segments that are identical to the TI and assigns higher

Fig. 7 Two images generated by a multi Gaussian model with correlation length of 20 and 50 in vertical and horizontal directions, respectively



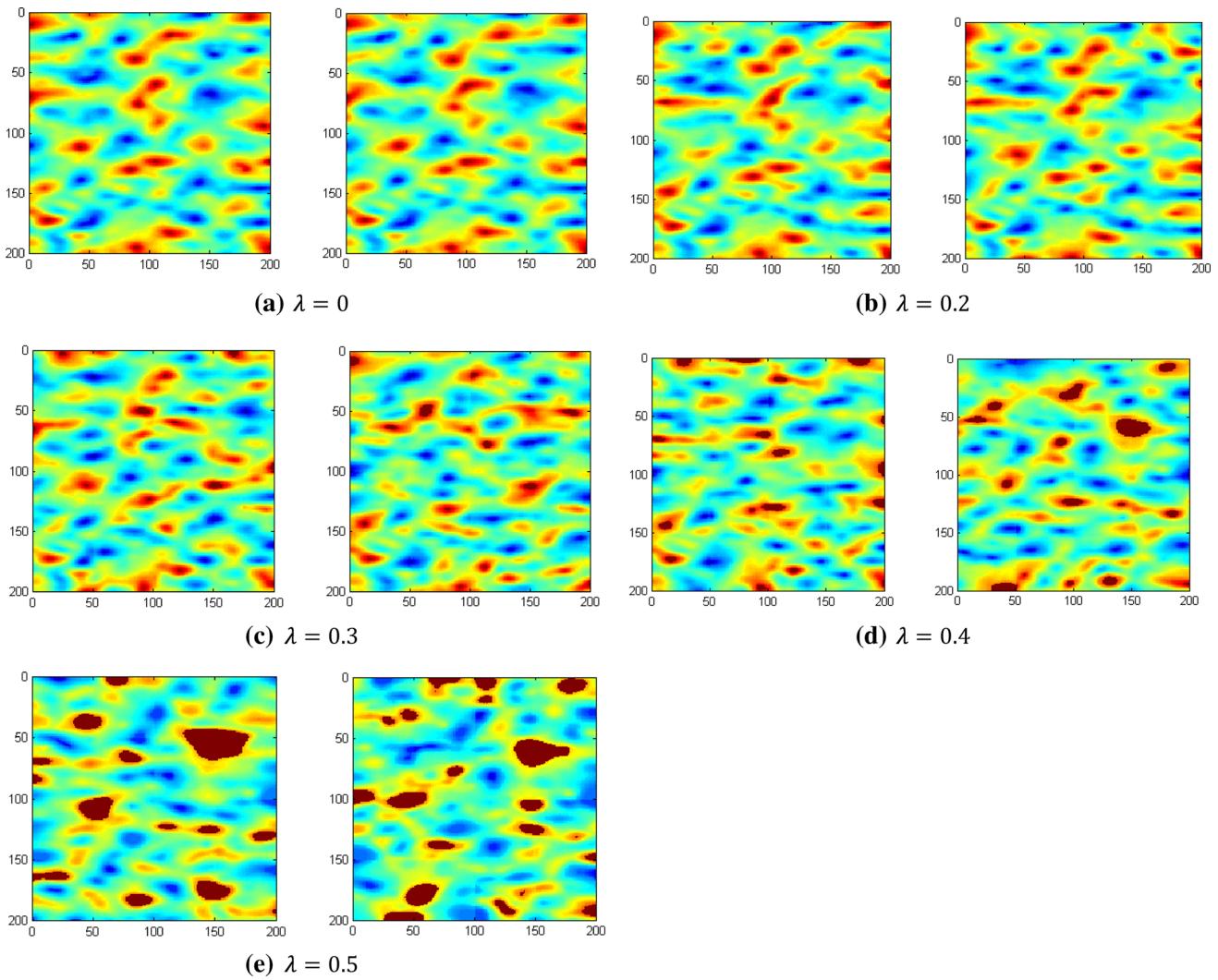


Fig. 8 Two samples of realizations generated by the proposed method for different λ

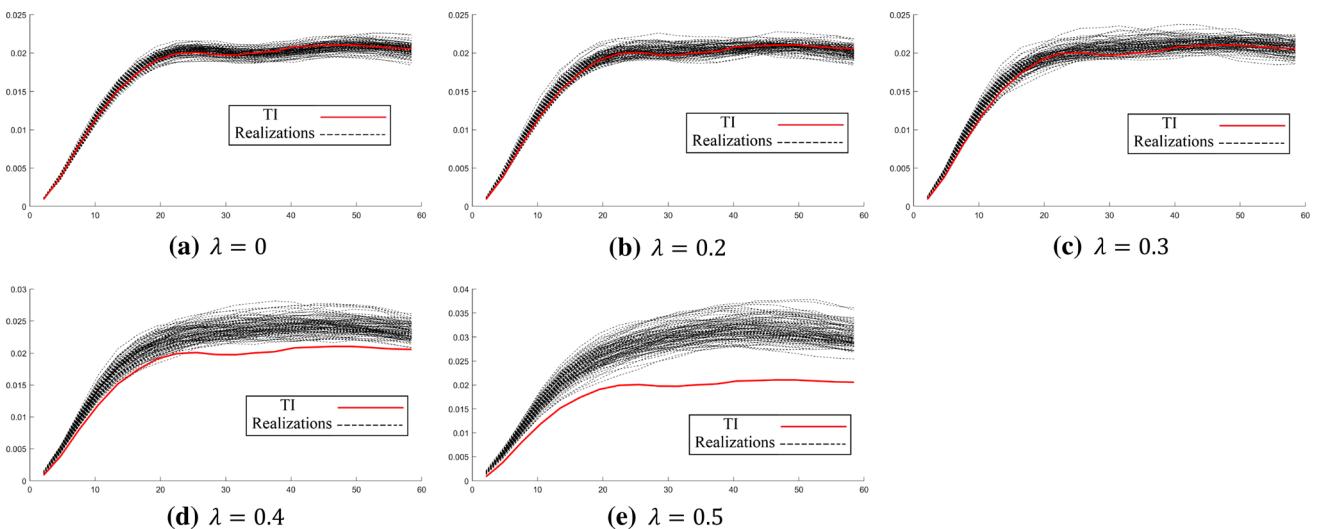


Fig. 9 each pane shows the variograms of 100 realizations obtained using the proposed method based on the TI and conditional data of Fig. 7 for different values of λ in black, along with the variogram of the TI in red. Increasing λ adversely affects the reproduction of the TI properties

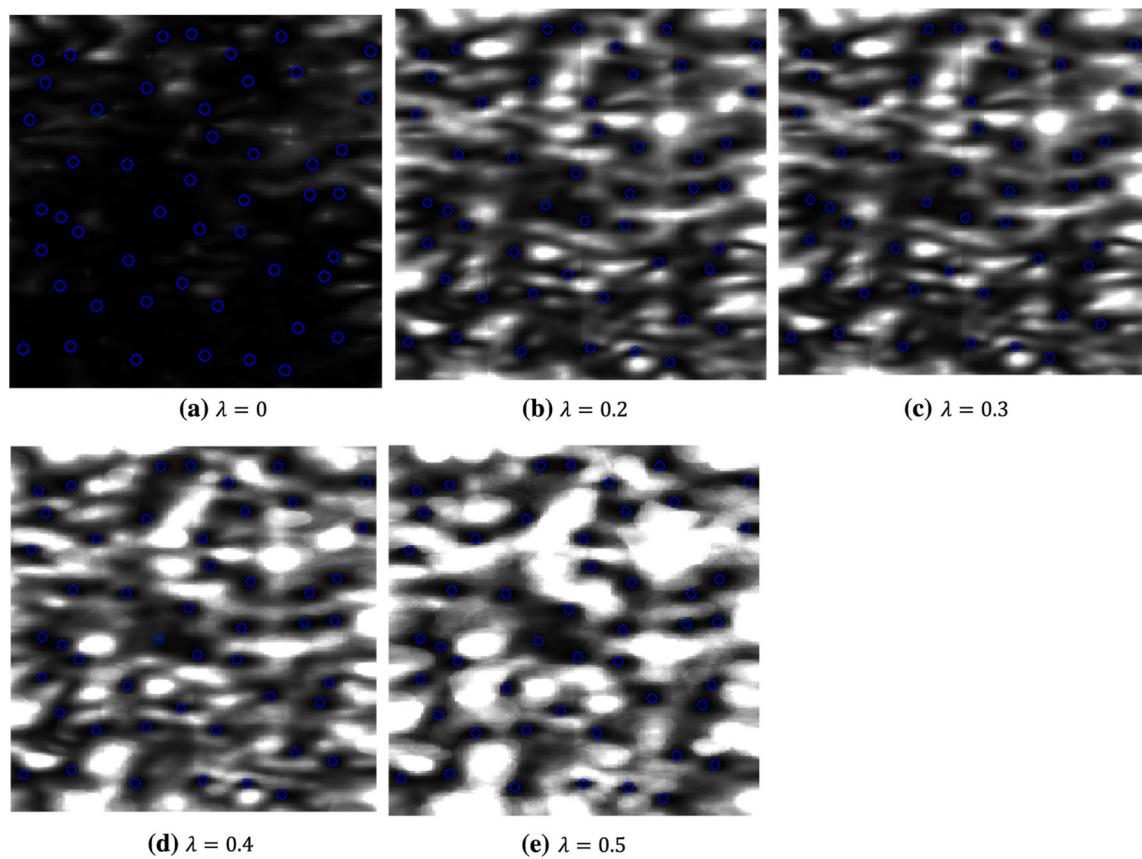


Fig. 10 Variance map of 100 realizations generated by proposed method. Local variance increases by increasing the λ , Location of samples are marked by blue circles

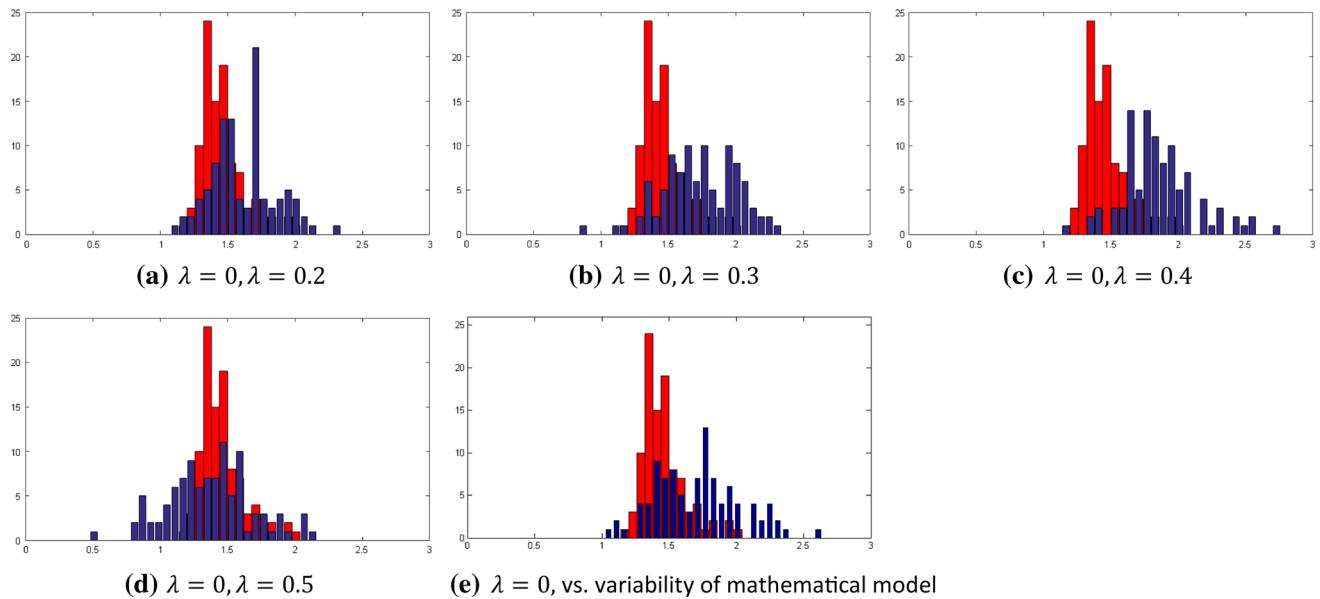


Fig. 11 Histograms of uphill effort in 100 realizations, red histograms generated by MPS and blue histograms are generated by different λ values, e shows the histogram of real fields generated through mathematical model

innovation to realizations that are composed of smaller segments.

For comparing two sets of realizations, $\gamma_i^{A,B}$ is defined as the relative innovation measure of set A , in comparison with set B :

$$\gamma_i^{A,B} = \frac{\sum_{k=1}^K (1 - C_{i,k}^B)}{\sum_{k=1}^K (1 - C_{i,k}^A)}, \quad (12)$$

where $C_{i,k}^A$ and $C_{i,k}^B$ is the innovation index of the k th realization in sets A and B , respectively (C_i lies in the interval $[0,1]$). If $\gamma_i^{A,B} > 1$, it indicates that set A contains more innovation than set B . Likewise, the inconsistency (or divergence) between a realization and the TI is shown with d and the relative innovation measure is defined as follows:

$$\gamma_c^{A,B} = \frac{\sum_{k=1}^K d_{c,k}^B}{\sum_{k=1}^K d_{c,k}^A} \quad (13)$$

$d_{c,k}$ shows the Jensen–Shannon divergence of the LBP histograms of the k th realization with the TI and lies in the interval $[0,1]$. $\gamma_c^{A,B} > 1$ shows that the realizations of set A are more consistent with the TI than those of set B .

Let us show the relative innovation and consistency factors for realizations obtained using the proposed method for $\lambda = \lambda_1$ with respect to similar realizations for $\lambda = \lambda_2$ with $\gamma_i^{\lambda_1, \lambda_2}$ and $\gamma_c^{\lambda_1, \lambda_2}$, respectively. First, we compute the vectors

$$\boldsymbol{\gamma}_i = [\gamma_i^{0,0}, \gamma_i^{0,1,0}, \gamma_i^{0,2,0}, \gamma_i^{0,3,0}, \gamma_i^{0,4,0}, \gamma_i^{0,5,0}]$$

and

$$\boldsymbol{\gamma}_c = [\gamma_c^{0,0}, \gamma_c^{0,1,0}, \gamma_c^{0,2,0}, \gamma_c^{0,3,0}, \gamma_c^{0,4,0}, \gamma_c^{0,5,0}].$$

These vectors are then normalized by linearly mapping them to the interval of $[0,1]$ and shown with symbols $\tilde{\gamma}_i$ and $\tilde{\gamma}_c$. $\tilde{\gamma}_i$ and $\tilde{\gamma}_c$ are expected to show monotonically increasing and decreasing values with λ . We modeled these variables by fitting a third order polynomial, denoted by f_i and f_c . The resulting curves for the realizations of Sect. 5.1 are depicted in Fig. 12.

Our goal is to choose λ so as to achieve the highest possible innovation while preserving consistency. One solution is to find the highest value of λ , denoted here by λ^* , that keeps normalized consistency score higher than a specific threshold t_c . For the realizations of Sect. 5.1, if we set $t_c = 0.8$, the corresponding λ^* is 0.2 as depicted in Fig. 12.

Setting $t_c = 0.8$ means that we allow the consistency factor to deviate for maximum of 20% from the most consistent realizations. In the following, we validate this choice by considering the uphill efforts obtained in the multi-Gaussian experiments. In this particular example, the

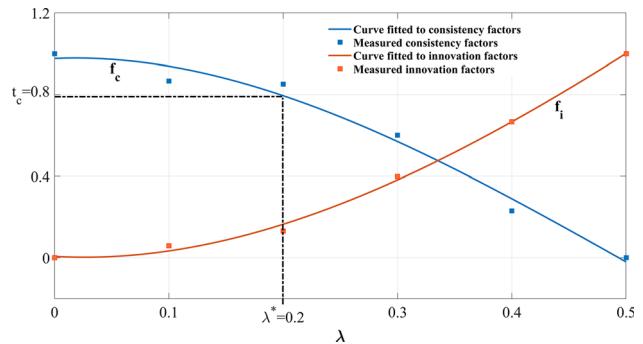


Fig. 12 For the realizations of Sect. 5.1, the innovation and consistency scores are computed for different λ and mapped to the interval of $[0,1]$. The best value for λ is the highest value which its normalized consistency score is higher than $t_c = 0.8$

true distribution of uphill efforts is known. A particular choice for λ can be evaluated by comparing the true distribution with the distribution obtained based on that λ . To compare two distributions, the Kullback–Leibler divergence is employed.

The distribution of uphill efforts obtained from the mathematical model is considered as the reference distribution, denoted \mathcal{P} . The probability distribution of the uphill efforts of the realizations obtained from the proposed method by the parameter λ is denoted by \mathcal{Q}_λ . Let us assume that the uphill efforts have normal distributions: $\mathcal{P} \sim \mathcal{N}(\mu, \sigma)$, $\mathcal{Q}_\lambda \sim \mathcal{N}(\mu_\lambda, \sigma_\lambda)$. The mean and variance values can easily be obtained from the computed efforts. The Kullback–Leibler divergence of two normal distributions can be computed as follows:

$$KL(\mathcal{P}, \mathcal{Q}_\lambda) = \log \frac{\sigma_\lambda}{\sigma} + \frac{\sigma^2 + (\mu - \mu_\lambda)^2}{2\sigma_\lambda^2} - \frac{1}{2}. \quad (14)$$

For the realizations of Sect. 5.1, the KL divergence is 3.75, 0.41, 0.16, 0.23, 0.07, and 0.66 for $\lambda = 0, 0.1, 0.2, 0.3, 0.4$ and 0.5, respectively. Based on these results, we can conclude that setting $\lambda^* = 0.2$ is a good choice. Although setting λ to 0.4 leads to a lower divergence, both visual judgment and consistency scores show that $\lambda = 0.4$ is not an appropriate setting.

We repeated the same test on the realizations of Fig. 8. The normalized consistency scores are then 1.0, 0.94, 0.83, 0.29 and 0 for $\lambda = 0, 0.2, 0.3, 0.4$ and 0.5, respectively. Considering $t_c = 0.8$, the selected value for λ should be approximately 0.3. Furthermore, the KL-divergence computed for the same λ values is 8.80, 0.24, 0.04, 0.16 and 0.65, showing that the lowest divergence occurs at $\lambda = 0.3$.

5.4 Tests on a TI presenting complex patterns

The distinguished strength of MPS simulation methods, as compared to two-point ones, is their capability of

Fig. 13 Lena Delta satellite imagery in Russia that is used as TI. **a** small TI (256×256). **b** Large TI (346×346)

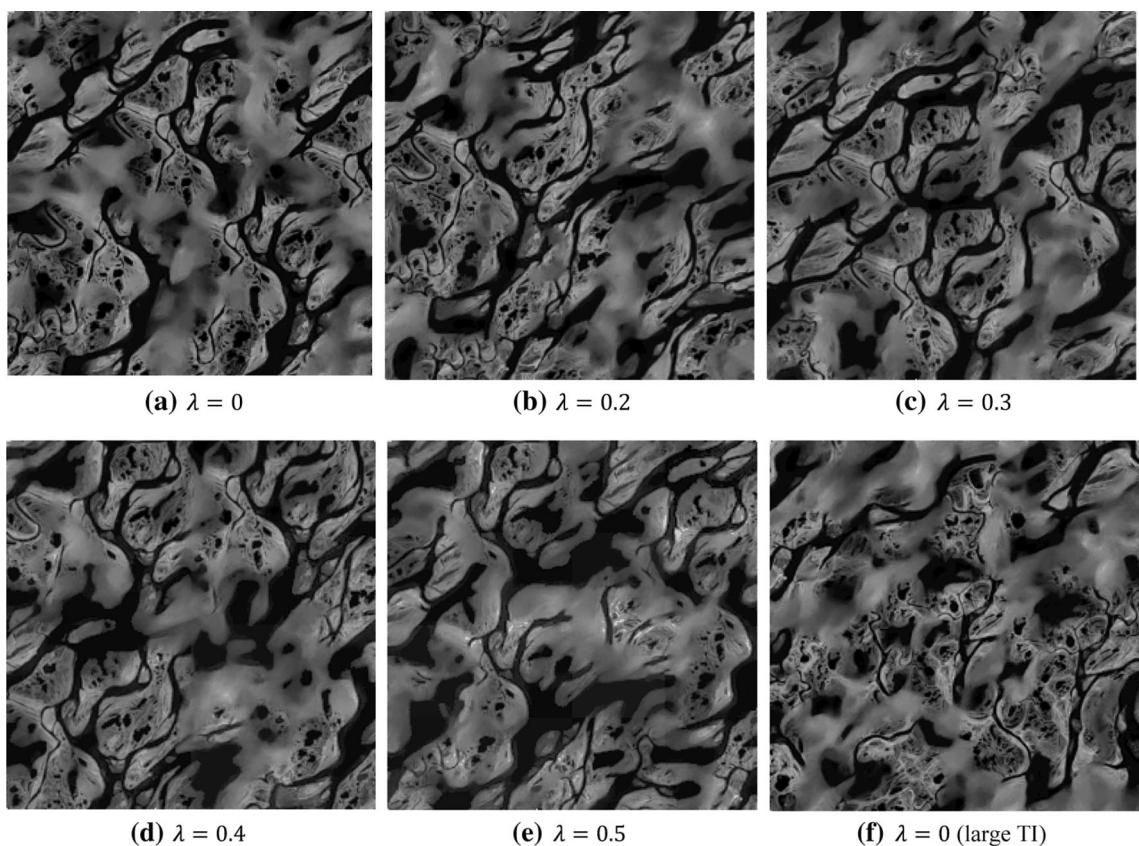
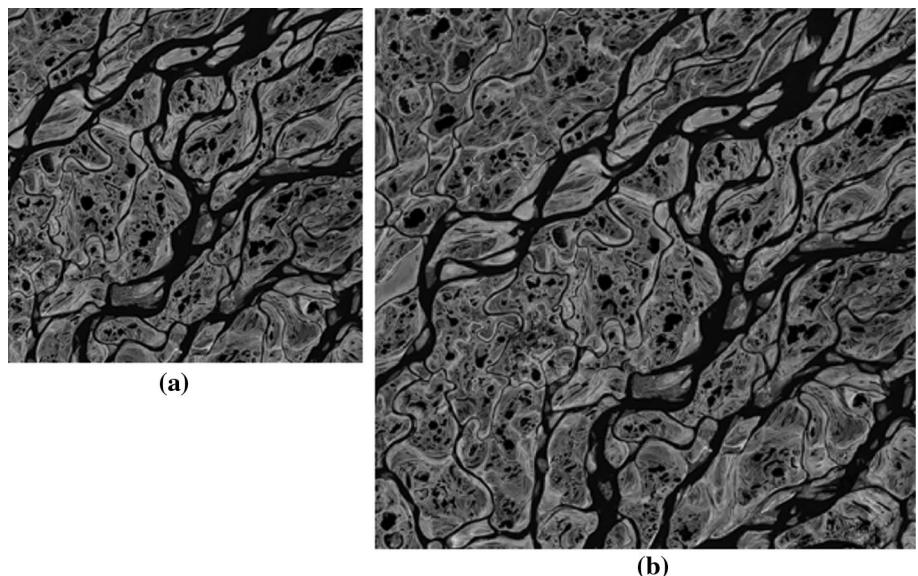


Fig. 14 Each figure shows one sample realization. **a–e** Are formed based the small TI (Fig. 10a) and the large TI of Fig. 10b is used for **f**. λ is 0, 0.2, 0.3, 0.4, 0.5, and 0 for **a–f**, respectively

reproducing complex and structured textures. In this section, our goal is to evaluate the proposed method on a real field with complex structures. Unfortunately, in the case of real images the mathematical or physical model of the generating process is usually unknown and hence, the

actual variation of the field cannot be determined. However, we can check the effect of the proposed method on increasing the local variances. Images shown in Fig. 13 are parts of *Lena Delta* satellite imagery in *Russia*. We have

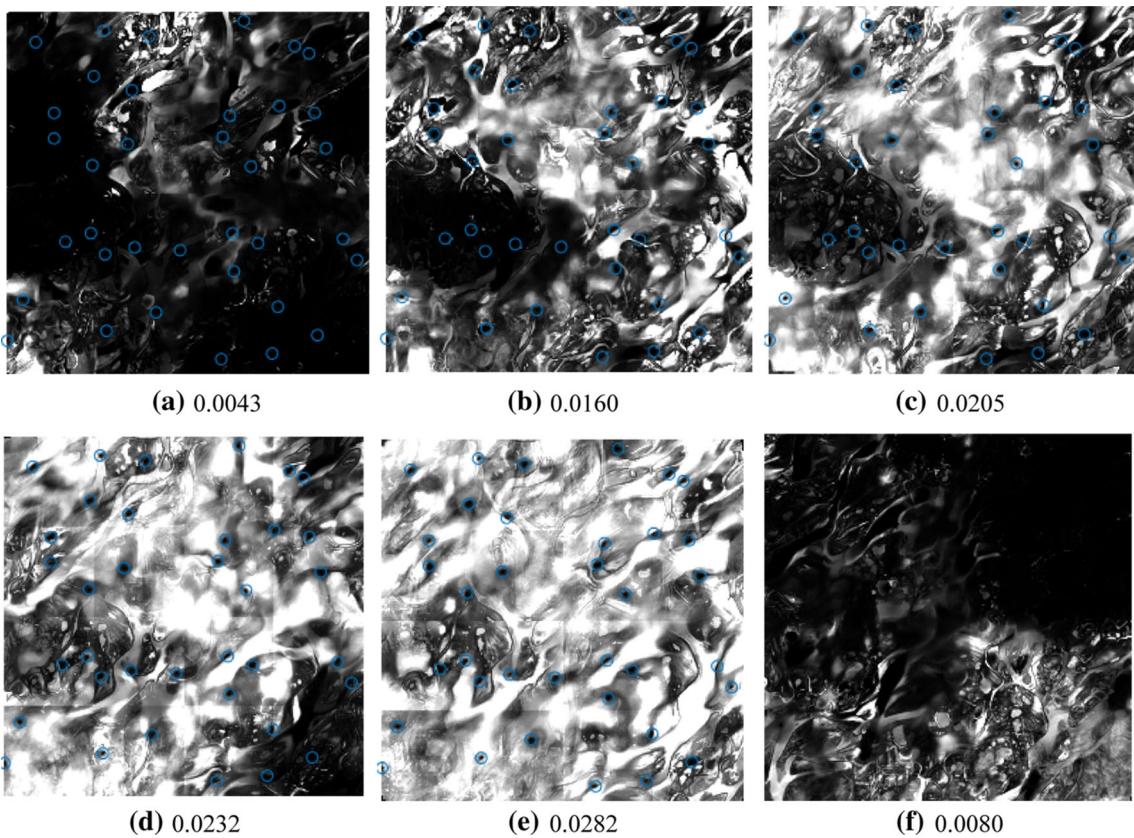
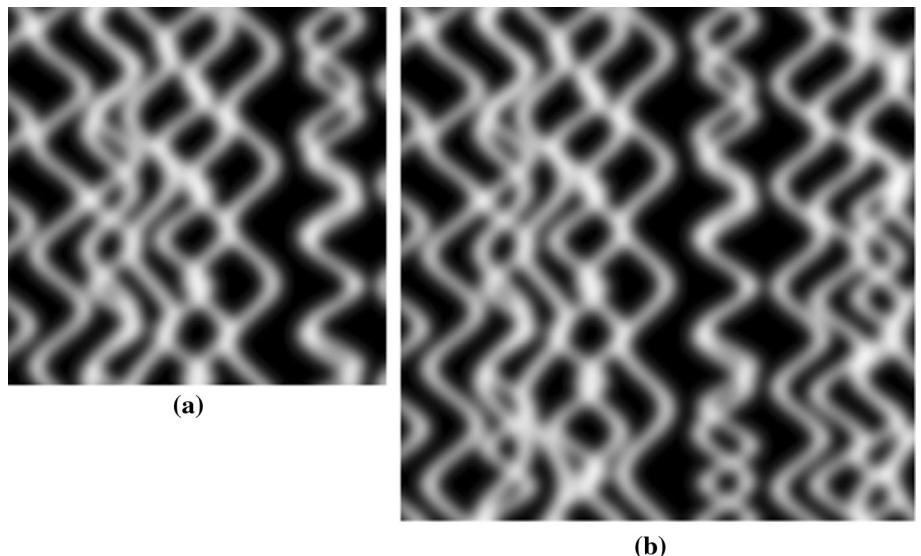


Fig. 15 Each figure shows the variance map of 100 realizations. The number of conditioning data is 36 and their location is marked with blue circles. **a–e** Are formed based the small TI (Fig. 10a) and the

large TI of Fig. 10b is used for \mathbf{f} . λ is 0, 0.2, 0.3, 0.4, 0.5, and 0 for $\mathbf{a-f}$, respectively. The mean variance is reported below each map

Fig. 16 Two synthetic TIs composed of sinusoidal channels. Pixels values are real values in the interval of [0,1].
a Small TI (256×256).
b Large TI (346×346)



used image a as a small training image and image b as a large one (notice that image a is a sub-image of b).

In order to generate samples, 36 values are taken randomly from the training image of Fig. 10a and placed in randomized locations in a simulation grid with the same

size. Then, 100 realizations are generated for each value of λ . The TIs used in this example contain very detailed structures and most simulation algorithms cannot perform well in reproducing them. The reason is that when searching for an acceptable match for a given patch in a

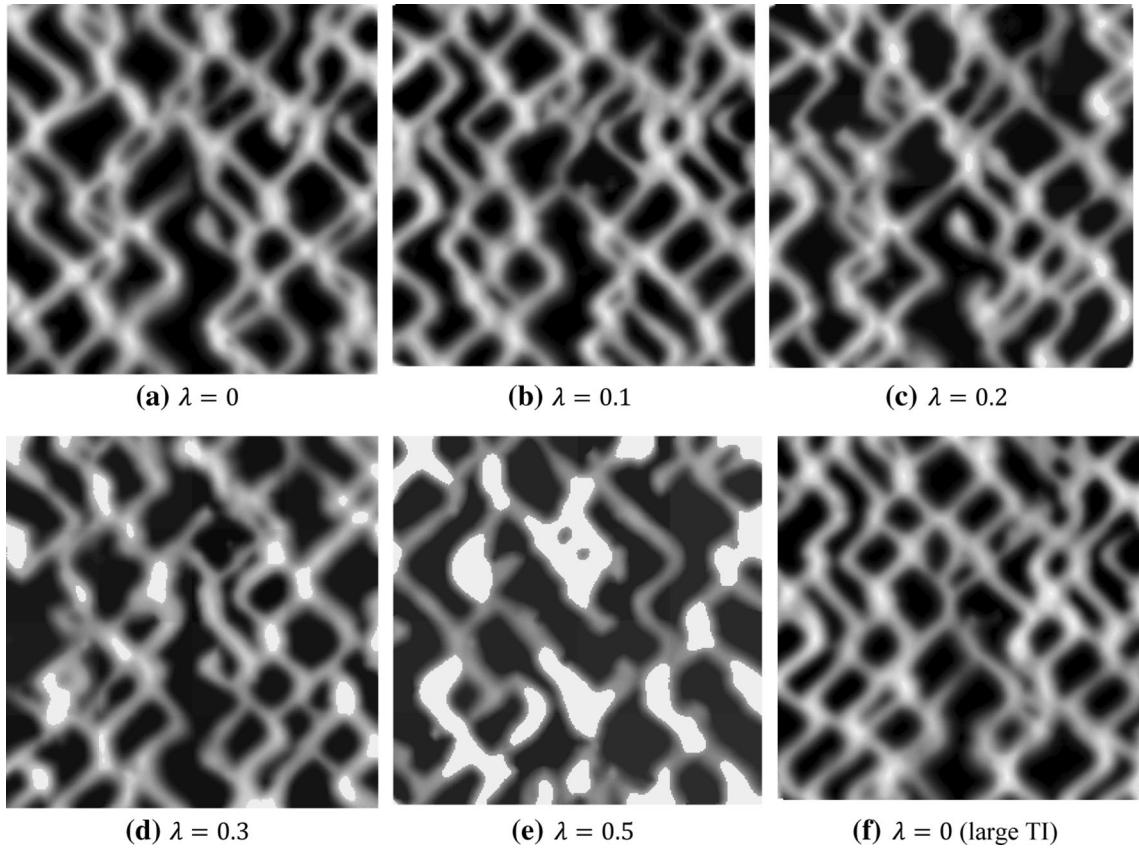


Fig. 17 Each figure shows one sample realization. **a–e** Are formed based the small TI (Fig. 16a) and the large TI of Fig. 16b is used for **f**. λ is 0, 0.1, 0.2, 0.3, 0.5, and 0 for **a–f**, respectively

particular overlap region, it is very rarely possible to find a good fit (other than its neighboring patches in the TI) and hence either the seams between different patches are always visible or large verbatim copies will be formed in the realizations. Some sample realizations obtained based on the proposed method for different λ are shown in Fig. 14. The problem between bad matches is somehow handled by blurring the overlap regions. This happens because of the averaging (expectation) phase in the optimization algorithm.

The variance maps of the realizations for different λ are shown in Fig. 15. As shown in the figure, the variance value for $\lambda = 0$ is close to zero even at unsampled locations. This means that the value of the field is almost the same in all different realizations. The reason is that the TI is very complex and includes diverse patterns, and hence very limited number of training patterns match with hard conditional data included in data-events. This problem almost is resolved by increasing λ .

The realizations used for computing all variance maps are simulated based on the small TI of Fig. 10a. The only exception is Fig. 14f (and equivalently Fig. 15f) which is formed based on the large TI for $\lambda = 0$. By comparing Fig. 15a, f, one can confirm that the mean of the variance

map has increased slightly by enlarging the TI (the mean variance is reported below each map). Increasing the value of λ to 0.2, however, leads to a significant increase in average variance without imposing any additional computational cost.

5.5 Tests on a synthetic TI

The proposed method is particularly useful when the TI is formed by densely sampling from a field (e.g. Rasera et al. (2019)). In such problems, much more money and time need to be spent for forming a larger TI, and the proposed method can help in modeling the real variability using a smaller TI. Based on this, one may argue that the proposed method will not be effective if the TI is synthesized using object-based simulators (Maharaja 2008). To test the performance of the proposed method on such cases, two TIs are synthesized by superimposing a number of sinusoidal channels. As shown in Fig. 16, the TIs contain repeating patterns. The TIs sizes are the same as in the previous example.

A simulation grid is formed by assigning random values to 36 random locations as conditioning data. In this test, 100 conditional realizations are produced for each

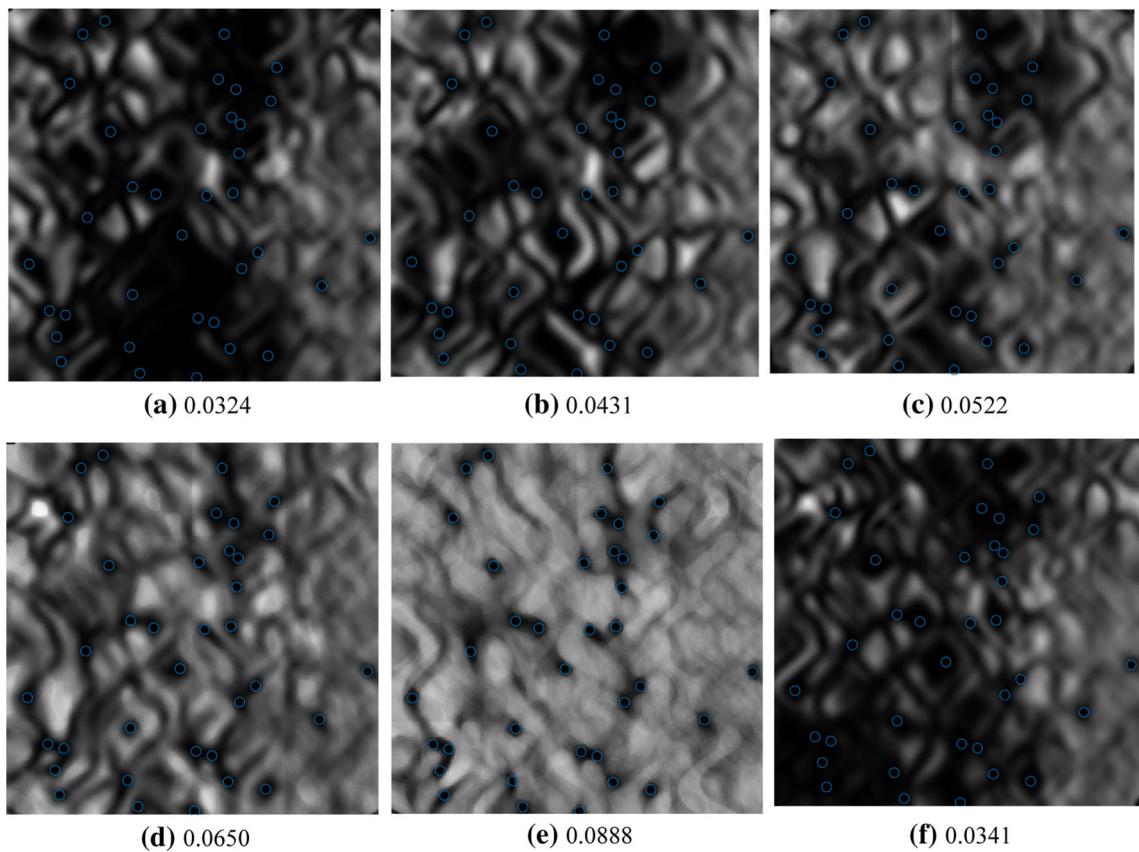


Fig. 18 Each figure shows the variance map of 100 realizations. The number of conditioning data is 36 and their location is marked with blue circles. **a–e** Are formed based the small TI (Fig. 16a) and the

large TI of Fig. 16b is used for **f**. λ is 0, 0.1, 0.2, 0.3, 0.5, and 0 for **a–f**, respectively. The mean variance is reported below each map

$\lambda \in \{0, 0.1, 0.2, 0.3, 0.5\}$. Example realizations for each setting are shown in Fig. 17. The realizations of Fig. 17a–e are synthesized based on the small TI, and the realization of Fig. 17f is simulated based on the large TI. Figure 18 shows the variance map for each setting and the average of the variance map is shown below each image. Average variance increases by increasing λ . Similar to the previous example, even though enlarging the TI increases the average variance, its influence is minimal compared to the effect of increasing λ .

Normalized consistency scores for $\lambda = 0, 0.1, 0.2, 0.3$ and 0.5 are computed as 1, 0.86, 0.54, 0.01, and 0 respectively, meaning that the best value for λ is about 0.1. Higher values for λ result in a significant decrease in the quality of the realizations as confirmed by the consistency scores and the visual judgment (see Fig. 17).

This experiment shows that, even for synthetic images, the proposed algorithm can increase the diversity of the realizations without reducing the quality of the realizations, if λ is appropriately selected. In such cases, the user can employ object-based simulators for synthesizing larger TIs to increase the variability. However, doing so requires much larger TIs and hence leads to a significant increase in

the computational cost. Therefore, in such cases the advantage of the proposed method is only to reduce the computational cost.

6 Conclusion

In this research, a novel method for increasing the variance in MPS problems based on optimization of an energy function is proposed. In this energy function, in addition to common important and necessary terms attempting to guarantee conditional data satisfaction and consistency with the training image, the goal of increasing the local variance is also included. By solving the obtained equations and considering the variance maximization constraint, a novel algorithm for MPS simulation is proposed.

Experiments show that the proposed algorithm can effectively compensate the problems caused by the limitedness of training patterns and facilitate access to realistic variability. This issue is particularly important in cases where large TIs are not available due to lack of sufficient data. Even if sufficiently large TIs are available, handling them in a MPS framework may result in

intractable computational expenses. For complicated textures, a TI that includes all possible variations of the texture will probably be gigantic and hence, intractable.

In many applications the purpose of geostatistical simulations is to investigate extreme cases. For example, when leakage of toxic substances occurs, we want to know how long it takes for these substances to contaminate a drinking water source, in order to plan remediation measures. The use of limited-sized training images reduces the possibility of investigating such extreme cases and underestimates the real diversity of the field. The proposed method effectively reduces this problem. The algorithm is based on parameter λ , which is a number between zero and one and the diversity of realizations increases with increasing λ . It should be noted that its excessive increase can lead to instability of the simulation and the generation of realizations with inappropriate structures. The experiments show that values in the range 0.1–0.3 for this parameter can significantly increase the variance of simulations and prevent instability at the same time. We have developed our variance maximization idea on a particular optimization-based MPS simulation algorithm. However, this goal can be integrated in other optimization-based, iterative or even sequential methods. This could be achieved by simulating a bunch of realizations simultaneously (and for sequential methods in a similar path) with a constraint to avoid similar patterns at the same location in different realizations.

Acknowledgements The authors would like to express their sincere thanks to anonymous reviewers who devoted their time and expertise to improving this paper.

References

- Abdollahifard MJ (2016) Fast multiple-point simulation using a data-driven path and an efficient gradient-based search. *Comput Geosci* 86:64–74. <https://doi.org/10.1016/j.cageo.2015.10.010>
- Abdollahifard MJ, Ahmadi S (2016) Reconstruction of binary geological images using analytical edge and object models. *Comput Geosci* 89:239–251
- Abdollahifard MJ, Faez K (2013) Stochastic simulation of patterns using Bayesian pattern modeling. *Comput Geosci* 17:99–116
- Abdollahifard MJ, Faez K (2014) Fast direct sampling for multiple-point stochastic simulation. *Arab J Geosci* 7:1927–1939
- Abdollahifard MJ, Mariethoz G, Ghavim M (2019) Quantitative evaluation of multiple-point simulations using image segmentation and texture descriptors. *Comput Geosci* 23:1349–1368
- Abdollahifard MJ, Mariethoz G, Pourfard M (2016) Improving in situ data acquisition using training images and a Bayesian mixture model. *Comput Geosci* 91:49–63
- Abdollahifard MJ, Nasiri B (2017) Exploiting transformation-domain sparsity for fast query in multiple-point geostatistics. *Comput Geosci* 21:289–299
- Arpat GB, Caers J (2007) Conditional simulation with patterns. *Math Geol* 39:177–203
- Emery X, Lantuéjoul C (2014) Can a training image be a substitute for a random field model? *Math Geosci* 46:133–147
- Honarkhah M, Caers J (2010) Stochastic simulation of patterns using distance-based pattern modeling. *Math Geosci* 42:487–517
- Hu L, Chugunova T (2008) Multiple-point geostatistics for modeling subsurface heterogeneity: a comprehensive review. *Water Resour Res*. <https://doi.org/10.1029/2008wr006993>
- Kalantari S, Abdollahifard MJ (2016) Optimization-based multiple-point geostatistics: a sparse way. *Comput Geosci* 95:85–98
- Kopf J, Fu C-W, Cohen-Or D, Deussen O, Lischinski D, Wong T-T (2007) Solid texture synthesis from 2d exemplars. In: ACM transactions on graphics (TOG), vol 3. ACM, p 2
- Kwatra V, Essa I, Bobick A, Kwatra N (2005) Texture optimization for example-based synthesis. In: ACM transactions on graphics (TOG), vol 3. ACM, pp 795–802
- Le Ravalec M, Noetinger B, Hu LY (2000) The FFT moving average (FFT-MA) generator: an efficient numerical method for generating and conditioning Gaussian simulations. *Math Geol* 32:701–723
- Li X, Mariethoz G, Lu D, Linde N (2016) Patch-based iterative conditional geostatistical simulation using graph cuts. *Water Resour Res*. <https://doi.org/10.1002/2015WR018378>
- Lyon JM, Ward TL (1980) Sequential calculation of the median. *Comput Ind Eng* 4:31–39
- Maharaja A (2008) TiGenerator: object-based training image generator. *Comput Geosci* 34:1753–1761
- Mahmud K, Mariethoz G, Caers J, Tahmasebi P, Baker A (2014) Simulation of Earth textures by conditional image quilting. *Water Resour Res* 50:3088–3107. <https://doi.org/10.1002/2013WR015069>
- Mariethoz G, Caers J (2014) Multiple-point geostatistics: stochastic modeling with training images. Wiley, Hoboken
- Mariethoz G, Renard P, Straubhaar J (2010) The direct sampling method to perform multiple-point geostatistical simulations. *Water Resour Res*. <https://doi.org/10.1029/2008wr007621>
- Oriani F, Straubhaar J, Renard P, Mariethoz G (2014) Simulation of rainfall time series from different climatic regions using the direct sampling technique. *Hydrol Earth Syst Sci* 18:3015–3031. <https://doi.org/10.5194/hess-18-3015-2014>
- Peredo O, Ortiz JM (2011) Parallel implementation of simulated annealing to reproduce multiple-point statistics. *Comput Geosci* 37:1110–1121
- Peyré G (2009) Sparse modeling of textures. *J Math Imaging Vis* 34:17–31
- Pourfard M, Abdollahifard MJ, Faez K, Motamed SA, Hosseini T (2017) PCTO-SIM: multiple-point geostatistical modeling using parallel conditional texture optimization. *Comput Geosci* 102:116–138
- Rasera LG, Gravey M, Lane SN, Mariethoz G (2019) Downscaling images with trends using multiple-point statistics simulation: an application to digital elevation models. *Math Geosci*. <https://doi.org/10.1007/s11004-019-09818-4>
- Rezaee H, Marcotte D, Tahmasebi P, Saucier A (2015) Multiple-point geostatistical simulation using enriched pattern databases. *Stoch Environ Res Risk Assess* 29:893–913
- Rezaee H, Mariethoz G, Koneshloo M, Asghari O (2013) Multiple-point geostatistical simulation using the bunch-pasting direct sampling method. *Comput Geosci* 54:293–308. <https://doi.org/10.1016/j.cageo.2013.01.020>
- Sharifzadehli M, Fathianpour N, Renard P, Amirkattahi R (2018) Random partitioning and adaptive filters for multiple-point stochastic simulation. *Stoch Environ Res Risk Assess* 32:1375–1396
- Strebelle S (2002) Conditional simulation of complex geological structures using multiple-point. *Stat Math Geol* 34:1–22
- Tahmasebi P (2018) Multiple point statistics: a review. In: Sagar BD, Cheng Q, Agterberg F (eds) *Handbook of mathematical geosciences; fifty years of IAMG*. Springer, Berlin, pp 613–643

- Tahmasebi P, Sahimi M (2016) Enhancing multiple-point geostatistical modeling: 2. Iterative simulation and multiple distance function. *Water Resour Res.* <https://doi.org/10.1002/2015WR017807>
- Yang L, Hou W, Cui C, Cui J (2016) GOSIM: a multi-scale iterative multiple-point statistics algorithm with global optimization. *Comput Geosci* 89:57–70

Zhang T, Switzer P, Journel A (2006) Filter-based classification of training image patterns for spatial simulation. *Math Geol* 38:63–80

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.