

# Addressing Conditioning Data in Multiple-Point Statistics Simulation Algorithms Based on a Multiple Grid Approach

Julien Straubhaar · Duccio Malinverni

Received: 18 December 2012 / Accepted: 22 June 2013 / Published online: 19 July 2013  
© International Association for Mathematical Geosciences 2013

**Abstract** Multiple-point statistics (MPS) allows simulations reproducing structures of a conceptual model given by a training image (TI) to be generated within a stochastic framework. In classical implementations, fixed search templates are used to retrieve the patterns from the TI. A multiple grid approach allows the large-scale structures present in the TI to be captured, while keeping the search template small. The technique consists in decomposing the simulation grid into several grid levels: One grid level is composed of each second node of the grid level one rank finer. Then each grid level is successively simulated by using the corresponding rescaled search template from the coarse level to the fine level (the simulation grid itself). For a conditional simulation, a basic method (as in *snesim*) to honor the hard data consists in assigning the data to the closest nodes of the current grid level before simulating it. In this paper, another method (implemented in *impala*) that consists in assigning the hard data to the closest nodes of the simulation grid (fine level), and then in spreading them up to the coarse grid by using simulations based on the MPS inferred from the TI is presented in detail. We study the effect of conditioning and show that the first method leads to systematic biases depending on the location of the conditioning data relative to the grid levels, whereas the second method allows for properly dealing with conditional simulations and a multiple grid approach.

**Keywords** Multiple-point statistics · Multiple grid approach · Conditional simulation

---

J. Straubhaar (✉)

The Centre for Hydrogeology and Geothermics (CHYN), University of Neuchâtel,  
Rue Emile-Argand 11, 2000 Neuchâtel, Switzerland  
e-mail: [julien.straubhaar@unine.ch](mailto:julien.straubhaar@unine.ch)

D. Malinverni

École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

## 1 Introduction

Multiple-point statistics (MPS) allows for powerful methods to generate complex geological fields in a stochastic framework. The basic principle of MPS methods is to use a conceptual model given by a training image (TI) displaying the wanted spatial features. As an application, the realizations of MPS algorithms can be set in input for flow or transport process (Huysmans and Dassargues 2012). MPS techniques are also widely used in inverse modeling where some geological conditioning data are available and some observations of physical processes have to be fit. For example, Ronayne et al. (2008) use MPS in combination with a probability perturbation method (Caers 2003; Caers and Hoffman 2006). The ensemble Kalman filter is employed for assimilating dynamic data and updating geostatistical simulations by Hu et al. (2013), Zhou et al. (2011) and Jafarpour and Khodabakhshi (2011). A frequency matching method is used by Lange et al. (2012), and Cordua et al. (2012), while Mariethoz et al. (2010a) and Alcolea and Renard (2010) propose to iteratively re-sample models for solving inverse problems.

MPS methods allow for integrating a conceptual model, but also has to be able to honor observed hard data. In MPS techniques, the conditioning strategy to hard data is a crucial point in order to accurately assess the uncertainty (variability). Special attention must be taken when simulating in neighborhoods of hard data, as the effects of a biased conditioning are most visible in those regions. Several algorithms have been developed since Guardiano and Srivastava (1993) introduced the concept of MPS, and each of them addresses the data conditioning. The original MPS algorithm *snesim* developed by Strebelle (2002) and the more recent algorithm *impala* proposed by Straubhaar et al. (2011) use a multiple grid approach (Tran 1994), and then have to deal with the data conditioning when the conditioning points do not fall on the coarse grid level, because the methods consist in a pixel-based sequential simulation starting with that level. In *snesim*, the hard data are relocated on the current grid level before simulating it. Accuracy of data conditioning in *snesim* and its impact on flow modeling are addressed in Saripally and Caers (2008). Other MPS techniques consist in sequentially patching patterns provided by the TI, such as in the *simpat* (Arpat and Caers 2007), *filtersim* (Zhang et al. 2006; Wu et al. 2008) or *ccsim* (Tahmasebi et al. 2012) algorithms; at each step of the simulation, the patched pattern has to honor the hard data. In *simpat*, the pattern is selected among a pattern data base in two steps by distinguishing the hard data from the previous simulated nodes, whereas in *filtersim* weighted distances giving more importance to the hard data are used. Whereas both algorithms also use a multiple grid approach, *ccsim* is based on a raster simulation path. This latter algorithm uses a cross-covariance function to express the similarity of patterns, and deals with the hard data by splitting the pattern into smaller regions. Unlike these previous methods, the direct sampling algorithm (Mariethoz et al. 2010b) straightforwardly addresses the conditioning data. Indeed, the method consists of a pixel-based sequential simulation where the TI is directly sampled and an adaptive search neighborhood is used during the simulation. Thereby, no statistics database, nor multiple grids are needed, and the hard data are simply assigned into the simulation grid before starting the simulation.

In this paper, a method for properly dealing with hard data in classical MPS algorithms using a multiple grid approach is proposed. This technique consists in spreading the conditioning data into each multiple grid level by simulation using the statistics inferred from the TI. The method is implemented in *impala* (Straubhaar et al. 2011; Straubhaar et al. 2013), which also allows for facies connectivity data conditioning (Renard et al. 2011). This paper is organized as follows. In Sect. 2, the principles of classical MPS algorithms for unconditional simulations and the concept of multiple grids are recalled. Then the conditional simulations in Sect. 3 are addressed, where the proposed method is explained in detail (Sect. 3.1), and the method based on relocation used in *snesim* is given (Sect. 3.2). Several tests showing the efficiency of the proposed method in comparison with the relocating strategy are presented in Sects. 4 and 5.

## 2 Multiple-Point Statistics (MPS) Algorithm Based on the Multiple Grid Approach

In this section, some notations related to MPS algorithms and the multiple grid approach are introduced.

### 2.1 Sequential Pixel-based MPS Simulation

A simulation grid  $G$  is populated with a categorical variable  $s$  by performing a sequential pixel-based simulation using statistics inferred from a TI. Assume that the facies codes are  $s = 0, \dots, M - 1$  and that  $s = -1$  is used as an uninformed value. A search template  $\tau$  is defined as a set of relative node locations (offsets)  $h_1, \dots, h_N$ ,  $1 \leq i \leq N$ . A facies at a node  $u$  is simulated as follows: (1) the data event  $d(u) = \{s(u + h_1), \dots, s(u + h_N)\}$  is retrieved from the simulation grid  $G$ , (2) the conditional probability distribution function (CPDF)

$$\mathbb{P}(s(u) = k | d(u)) = \frac{\#\{v \in TI : s(v) = k \text{ and } s(v + h) = s(u + h) \forall h \in \tau \text{ s.t. } s(u + h) \neq -1\}}{\#\{v \in TI : s(v + h) = s(u + h) \forall h \in \tau \text{ s.t. } s(u + h) \neq -1\}}, \quad (1)$$

is computed, and (3) used for randomly drawing a facies code at  $u$ . Note that the further informed node is dropped in  $d(u)$  until the denominator of Eq. (1) does not vanish. Thus, for an unconditional simulation, all nodes in  $G$  are initialized by setting the value  $-1$ , and each node is then successively simulated, conditional to the previous simulated nodes within a given neighborhood. Note that the computation of the CPDF (1) is done using a catalog storing the data events found in the TI, which is scanned only once before the simulation. This catalog is managed with a tree structure in *snesim* (Strebelle 2002) and with both list and tree structures in *impala* (Straubhaar et al. 2013).

## 2.2 Multiple Grid Approach

The multiple grid approach introduced by Tran (1994) allows for capturing the large-scale structures within the TI while keeping the search template size small. It consists in considering subgrids at different scales in the simulation grid, and simulating the nodes from the coarse scale to the fine scale level by using a search template rescaled accordingly. More precisely, a simulation grid is a boxed-shaped set of pixels with integer coordinates

$$G = \{(n_x, n_y, n_z) : 0 \leq n_x \leq N_x - 1, 0 \leq n_y \leq N_y - 1, 0 \leq n_z \leq N_z - 1\}, \quad (2)$$

where  $N_x$ ,  $N_y$ , and  $N_z$  are the dimensions along the  $x$ -axis,  $y$ -axis, and  $z$ -axis, respectively. Note that for bi-dimensional simulations,  $N_z$  is set to 1. A subgrid is defined by the offsets (integer lags)  $l_x$ ,  $l_y$ , and  $l_z$  between two neighboring nodes along each direction

$$SG(l_x, l_y, l_z) = \{(n_x, n_y, n_z) \in G : n_x \equiv 0 \pmod{l_x}, n_y \equiv 0 \pmod{l_y}, n_z \equiv 0 \pmod{l_z}\}. \quad (3)$$

For MPS simulations, a number  $m$  of multiple grid levels is defined, and the subgrids

$$SG_i = SG(2^{m-i}, 2^{m-i}, 2^{m-i}), \quad 1 \leq i \leq m, \quad (4)$$

are considered. Thus, we have  $SG_1 \subset SG_2 \subset \dots \subset SG_m = G$ , and the sets

$$MG_1 = SG_1, \quad MG_i = SG_i \setminus SG_{i-1}, \quad 1 < i \leq m, \quad (5)$$

called the *multigrids*, form a partition of the simulation grid  $G$ , and verify the relations

$$SG_i = \bigcup_{j=1}^i MG_j, \quad 1 \leq i \leq m. \quad (6)$$

The simulation proceeds by populating the pixels of the multigrids  $MG_1, \dots, MG_m$  successively, that is, from the coarse level to the fine level. The simulation of the nodes in the multigrid  $MG_i$ , that is the uninformed nodes in the subgrid  $SG_i$ , is performed along a random path, by using the rescaled search template

$$\tau_i = \{2^{m-i} \cdot h_1, \dots, 2^{m-i} \cdot h_N\}, \quad (7)$$

where  $\tau = \{h_1, \dots, h_N\}$  is the initial search template defined on  $G$ .

Following this approach, the structures at large scale are simulated first, by using a search template that covers a wide area while keeping its size small. Note that a different initial search template  $\tau$  (before rescaling) can be specified for each multiple grid level. Indeed, since the fine multigrid contains the major part of the simulation grid (about 75 % and 87.5 % for bi- and three-dimensional cases, respectively), choosing templates of decreasing size from the coarse to the fine level allows a lot of computational time to be saved. This does not significantly affect

the quality of the realizations, because when simulating the fine levels, the structures at larger scales are already simulated and guide the simulation. Moreover, note that distinct numbers  $m_x, m_y, m_z$  of levels can be defined for each axis direction. In this situation, we set  $m = \max(m_x, m_y, m_z)$ , define the  $i$ th subgrid as  $SG_i = SG(\max(2^{m_x-i}, 1), \max(2^{m_y-i}, 1), \max(2^{m_z-i}, 1))$ , and rescale the lag vectors constituting the initial search template accordingly.

### 3 Conditional Simulations

In presence of hard data, we assume that the simulation grid covers every conditioning location (given by real coordinates) and that its resolution is fine enough so that two distinct conditioning points do not fall onto the same pixel. The grid node or pixel given by the integer coordinates  $(n_x, n_y, n_z)$  in  $G$  (defined by Eq. (2)) corresponds to a region  $[o_x + n_x \cdot d_x, o_x + (n_x + 1)d_x] \times [o_y + n_y \cdot d_y, o_y + (n_y + 1)d_y] \times [o_z + n_z \cdot d_z, o_z + (n_z + 1)d_z]$  in real space, where  $(o_x, o_y, o_z)$  is the origin and  $(d_x, d_y, d_z)$  the resolution of the simulation grid.

To honor the hard data and ensure the spatial continuity of the simulated features, the hard data must be taken into account during the simulation of every multigrid level. Then a strategy has to be developed to bring the information of the hard data up to the coarsest level. In the following, a method implemented in *impala* using MPS simulation is proposed in order to achieve this. Then we recall the method implemented in *snessim* consisting in relocating the hard data before the simulation of each subgrid.

#### 3.1 Spreading Hard Data by MPS Simulation

The idea of the proposed method for bringing the information of hard data into each multigrid level is to use MPS inferred from the TI to ensure consistency with the conceptual model. In a situation with  $m$  levels, the hard data is first assigned into the simulation grid  $G = SG_m$ , and the neighboring nodes of the hard data locations in the subgrid  $SG_{m-1}$  are simulated by using the template  $\tau_m$  associated to the scale of  $SG_m$ . Then we continue by simulating the neighboring nodes in  $SG_{m-2}$  with  $\tau_{m-1}$ , and so on. At each scale, the neighboring nodes around the hard data are simulated by using the template of the scale one rank finer. More precisely, the method consists in the following steps:

1. *Initializing the simulation grid.* The variable at each node of the simulation grid  $G$  is set to the uninformed value.
2. *Assigning hard data.* For each hard datum, its facies code is assigned to the closest node in the simulation grid  $G = SG_m$ . We note  $HD$  following the set of the informed node locations after this step.
3. *Spreading the hard data into the subgrids.* For  $i = m - 1, m - 2, \dots, 1$ , we successively do the following.
  - (a) We define a set of weak conditioning nodes  $WD_i$  in  $SG_i$  as follows. For a node location  $u$  in  $HD$ , we define  $W_i(u)$  as the set of nodes in the subgrid  $SG_i$

enclosing the node  $u$ , that is, if  $u = (u_x, u_y, u_z)$  are the integer coordinates of the pixel  $u$ , we set

$$u_x^- = \max\{v \in \mathbb{N} : v \equiv 0 \pmod{2^{m-i}} \text{ and } v \leq u_x\}, \quad (8)$$

$$u_x^+ = \min\{v \in \mathbb{N} : v \equiv 0 \pmod{2^{m-i}} \text{ and } v \geq u_x\}, \quad (9)$$

and  $u_y^-, u_y^+, u_z^-, u_z^+$  similarly, and define

$$W_i(u) = \{v = (v_x, v_y, v_z) \in SG_i : v_x \in \{u_x^-, u_x^+\}, v_y \in \{u_y^-, u_y^+\}, v_z \in \{u_z^-, u_z^+\}\}. \quad (10)$$

Then the set of weak conditioning nodes  $WD_i$  is defined as

$$WD_i = \bigcup_{u \in HD} W_i(u). \quad (11)$$

- (b) Then each uninformed node in  $WD_i$  is simulated using the search template  $\tau_{i+1}$ , that is, the template corresponding to the scale of the subgrid one rank finer,  $SG_{i+1}$ .

Finally, the simulation continues by populating all the uninformed nodes in the multigrids  $MG_1, MG_2, \dots, MG_m$  as described in Sect. 2. For a bi-dimensional simulation grid with 3 multigrid levels and one hard datum, any situation is similar to one of the six cases shown in Fig. 1. Note that no conflict occurs when more than one hard datum lead to a common weak conditioning node in the set resulting from the union (11) (step 3a), because the facies at the hard data are not considered: The weak conditioning nodes are the locations in a subgrid where a facies will be simulated (step 3b).

It is important to emphasize that no additional computation is required to build the catalogs of statistics used for simulating the weak conditioning data (step 3b above). Indeed, these statistics (one catalog per scale of subgrid/multigrid) are needed for conditional and unconditional simulations, and every catalog is built once at the beginning and stored in memory until the end of the execution of the program, which is conceivable through the use of the list structure, cheap in terms of memory use, and implemented in *impala*. Hence, this strategy for dealing with conditioning data is not penalizing in terms of computational time.

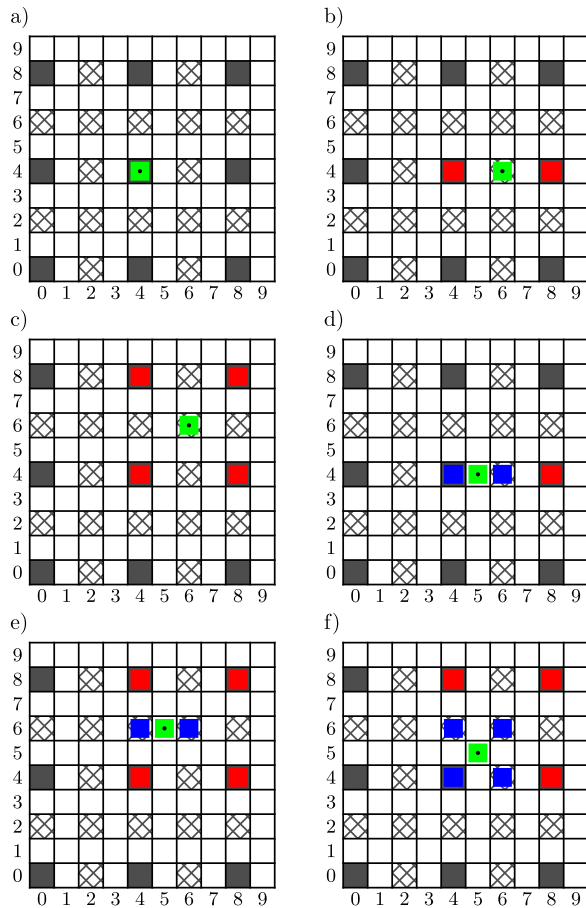
### 3.2 Relocating the Hard Data

A much simpler method for addressing the hard data conditioning is used in *snesim*. The main steps of the algorithm with  $m$  multiple grid levels for a conditional simulation are the following (Remy et al. 2009):

1. The simulation grid  $G$  is initialized by setting every node to the uninformed value;
2. For  $i = 1, \dots, m$ :
  - (a) Relocate every hard data to the nearest node in  $SG_i$ .
  - (b) Simulate every uninformed node in  $SG_i$ , by using the search template  $\tau_i$  (corresponding to the scale of this subgrid).
  - (c) Remove the relocated hard data from  $SG_i$  if  $i < m$ .

**Fig. 1** Spreading the hard data into all the subgrids.

A bi-dimensional simulation grid  $G$  with 3 multigrid levels is considered: gray nodes for coarse multigrid  $MG_1$ ; hatched nodes for middle multigrid  $MG_2$ ; white nodes for fine multigrid  $MG_3$ . Considering one hard datum, any situation is similar to one of the cases illustrated from (a) to (f): the *black point* marks the location of the hard datum; the *green square* is the corresponding hard data node in  $G = SG_3$ ; the *blue squares* are the nodes in  $WD_2$  simulated by using the search template  $\tau_3$ ; the *red squares* are the nodes in  $WD_1$  simulated by using the search template  $\tau_2$

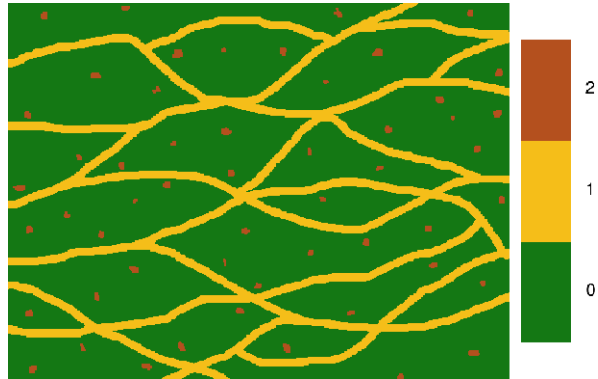


Relocating the hard data presents two drawbacks: (1) depending on their location and the current subgrid level, more than one datum can be relocated on the same subgrid node, and (2) following this strategy can be inconsistent with the spatial statistics inferred by the TI. The method is recalled here because it will be compared in the next sections to the proposed method presented in Sect. 3.1.

### 4 Illustrative Example

We consider the hand drawn bi-dimensional TI displayed on Fig. 2, of dimensions  $640 \times 480$  and containing 3 facies. The facies 1 represents channels, the facies 2 some small structures, and the facies 0 the matrix. We define 10 hard data in a simulation grid of size  $200 \times 200$ , and set the number of multiple grid levels to 5. Although this number is rather large compared to the dimensions of the simulation grid, it is chosen to show the effect of the multiple grid approach on the data conditioning, and it is justified by the large-scale structures depicted by the channels (facies 1). The

**Fig. 2** Training image used for the tests,  $640 \times 480$ , 3 facies



proposed method implemented in *impala* is compared to *snesim*: 500 realizations are generated by both algorithms and the results are shown in Fig. 3. One realization provided by each algorithm is displayed on Figs. 3a and 3b, and shows that the input parameters for the simulations give results of similar quality for both algorithms. The occurrence proportion maps for each facies and both algorithms are shown in Figs. 3c–3h. We observe that a wider area is frozen around the hard data when using *snesim* because of the relocation of the conditioning points in each subgrid level. This is studied in more detail in the next section.

## 5 Effect of Conditioning

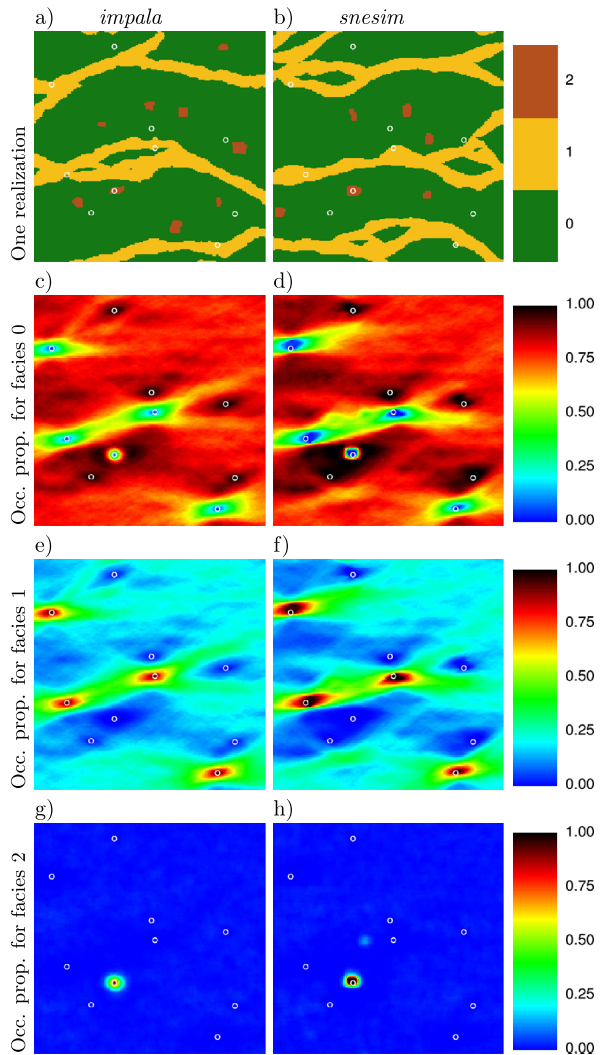
The purpose of this section is to demonstrate through examples that the proposed method properly addresses the hard data conditioning, that is, consistent with the statistics of the conceptual model. For the tests, the previous bi-dimensional TI (Fig. 2) is considered, because it contains large and small structures (facies 1 and 2, respectively) and will allow to observe the behavior of the conditioning in the MPS simulations when hard data values represent different features. For the first examples (Sect. 5.1), only one hard datum located around the center of the simulation grid is considered, and the variability (uncertainty) around the datum compared depending on its location relative to the multiple grid levels. Next (Sect. 5.2), we consider a dense hard data set and show that the method is stable when using different number of multiple grid levels.

### 5.1 Examples with a Unique Hard Datum

As in Sect. 4, the number of multiple grid levels is set to 5 and a  $200 \times 200$  simulation grid is considered. For every test in this section, a unique hard datum placed in the finest multigrid is considered. Several tests are performed with two different (but close) locations in the finest multigrid, because it represents the worst case in the sense that the information of the hard datum has to be spread through all the multiple grid levels. Since only one hard datum is considered, it allows for a comparison with the rejection method, which consists in generating unconditional simulations

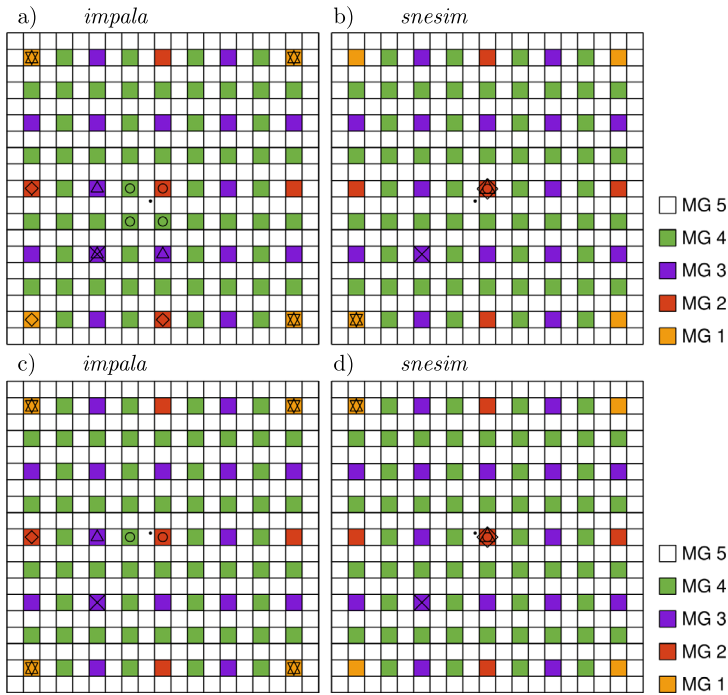


**Fig. 3** Tests with 10 hard data in a simulation grid of dimensions  $200 \times 200$ , using the TI in Fig. 2. Results using *impala* and *snesim* are displayed in the left and right columns respectively; (a–b) one realization; (c–h) occurrence proportion maps computed over 500 realizations for: (c–d) facies 0, (e–f) facies 1, (g–h) facies 2. The *whites circles* mark the hard data locations in each map



and retaining only those honoring the hard datum. For each case, 500 realizations are generated, using *impala*, *snesim* and the rejection method (based on *impala*).

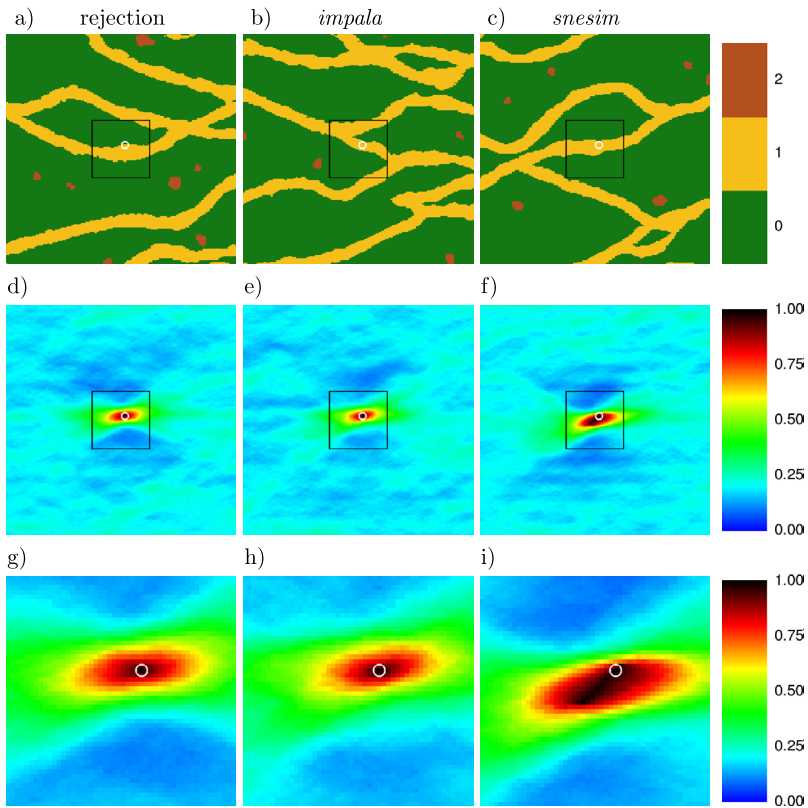
Figure 4 shows the conditioning strategies of *impala* (left column) and *snesim* (right column), according to the hard datum location given by the black dot. This location is near the center of the simulation grid and slightly differs between the top and bottom rows of Fig. 4. Note that the cross-marked node is the pixel of coordinates (100, 100) in the simulation grid  $G$ . For *impala* (Figs. 4a, 4c): the hard datum value is first assigned to the node containing it; then, the nodes marked by a circle are simulated using the template at the scale of  $MG_5$ , the triangle-marked nodes at the scale of  $MG_4$ , the diamond-marked nodes at the scale of  $MG_3$ , and the star-marked nodes at the scale of  $MG_2$ . Finally, the remaining uninformed nodes are simulated from the coarsest multigrid ( $MG_1$ ) to the finest one ( $MG_5$ ). For *snesim* (Figs. 4b, 4d),



**Fig. 4** Conditioning strategies in *impala* (left column) and *snesim* (right column), for two situations (top and bottom rows) with one hard datum. Each map is a part of the  $200 \times 200$  simulation grid around the pixel of coordinates (100, 100) marked by a cross. The black dot shows the location of the hard datum. Following the proposed strategy (*impala*): (a, c) circle-marked nodes: weak conditioning nodes in  $SG_4$  simulated at the scale of  $MG_5$ ; triangle-marked nodes: weak conditioning nodes in  $SG_3$  simulated at the scale of  $MG_4$ ; diamond-marked nodes: weak conditioning nodes in  $SG_2$  simulated at the scale of  $MG_3$ ; star-marked nodes: weak conditioning nodes in  $SG_1$  simulated at the scale of  $MG_2$ ; following the strategy implemented in *snesim*: (b, d) the star, diamond, triangle, and circle mark the relocation of the hard datum for the simulation at the scale of  $MG_1$ ,  $MG_2$ ,  $MG_3$ , and  $MG_4$ , respectively

the hard datum is first relocated to the star-marked node for the simulation of  $SG_1$ , then to the diamond-marked node for the simulation of uninformed nodes in  $SG_2$ , to the triangle-marked node for  $SG_3$ , to the circle-marked node for  $SG_2$ , and finally to the node containing the black dot for  $SG_5$ .

The results displayed on Figs. 5 and 6 are obtained with a hard datum value of 1 (channels) located in the simulation grid as shown respectively in the top and bottom rows of Fig. 4. As expected, the realizations in the first line of Figs. 5 and 6 show that a channel goes through the hard datum. Moreover, the results obtained by using the proposed method show that, on average, the location of this channel is centered on the hard datum (Figs. 5e, 5h, 6e, 6h), same as for the results provided by the rejection method (Figs. 5d, 5g, 6d, 6g). On the contrary, when using *snesim*, the hard datum lies on average on the border of the channel: the upper border (Figs. 5f, 5i) or the lower border (Figs. 6f, 6i), depending on how the hard datum is relocated onto the coarsest multiple grid level (Figs. 4b, 4d). The hard datum location between the two situations differs only from one pixel in the vertical direction, and then very few

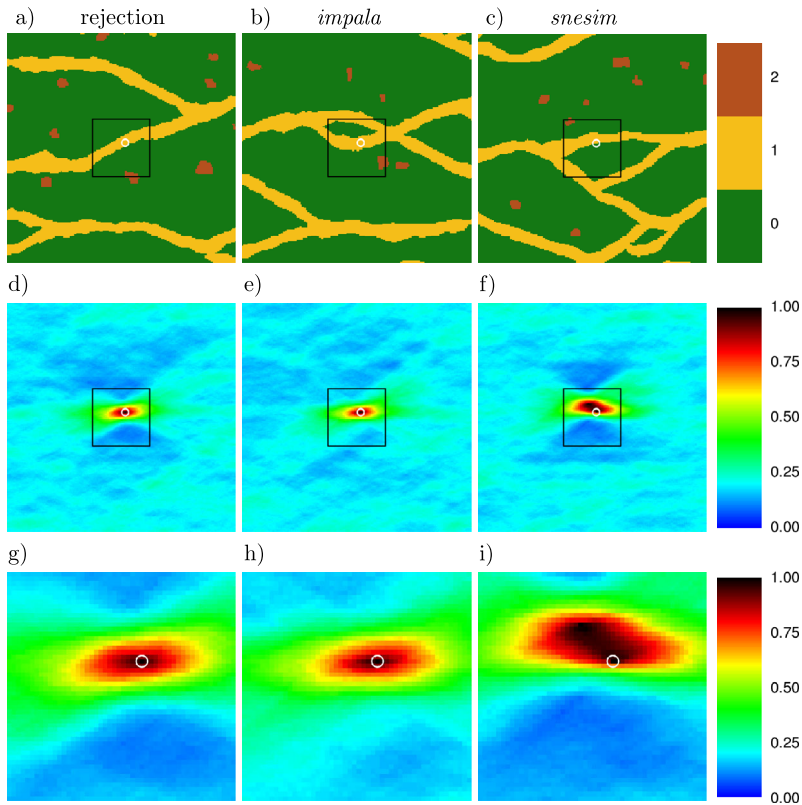


**Fig. 5** Results for the situation with one hard datum of value 1 (channel) located as shown in the top row of Fig. 4. The node containing the hard datum is highlighted with a white circle in every map. (Top row) One realization ( $200 \times 200$ ) obtained by (a) rejection method, (b) *impala* and (c) *snesim*; (middle row) occurrence proportion map for facies 1 for 500 realizations generated by (d) rejection method, (e) *impala* and (f) *snesim*; (bottom row) (g–i) zoom on the squared area displayed in the previous row

changes are expected locally on the occurrence proportion maps. This is the case for the rejection method and the proposed method (*impala*), which provide similar occurrence proportion maps, but not when *snesim* is used.

Similar tests are presented in Figs. 7 and 8, but setting the hard datum value to 2 (small structure). The above observations are corroborated by these examples.

Moreover, we compute the size of the geobody containing the hard datum, in number of nodes, for every realizations. The results are presented with boxplots in Fig. 9 and compared to the size of the geobodies constituted by the facies 2 found in the TI (Fig. 2). The rejection method and the proposed method (*impala*) give close results and the inter-quartile of the geobody sizes is included in the range of the geobody sizes from the TI, whereas relocating the hard datum (*snesim*) leads to a systematic bias; the geobody containing the hard datum covers a larger area, often exceeding the largest geobody (facies 2) found in the TI. This example emphasizes that the bias induced by the relocation strategy is significant when the sizes (diameters) of the typical structures are comparable to the node step (lag between two neighboring nodes)



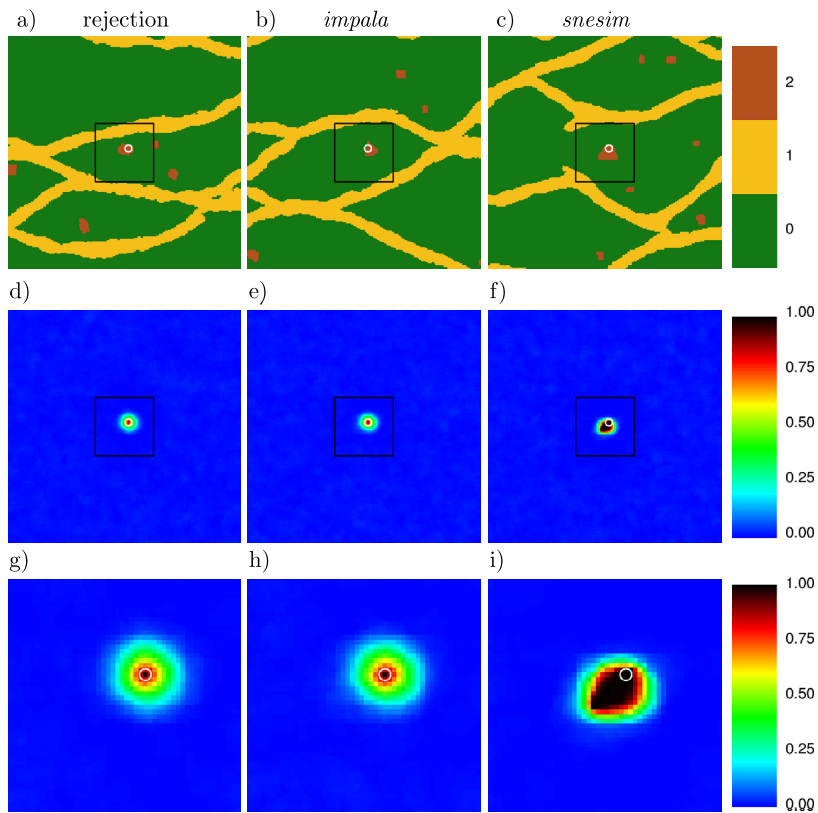
**Fig. 6** Results for the situation with one hard datum of value 1 (channel) located as shown in the bottom row of Fig. 4. The display is organized as in Fig. 5

in the coarsest subgrid. Indeed, in this example, the diameter of the geobodies constituted by the facies 2 found in the TI is not larger than  $2^4 = 16$ , which corresponds to the node step for the first of the 5 multiple grid levels.

## 5.2 Examples with a Dense Hard Data Set

A set of 200 hard data in a simulation grid of dimensions  $200 \times 200$ , randomly extracted from an unconditional simulation (reference image) is now considered. This is then compared with the results of 500 simulations by setting 3 and 5 multiple grid levels and using *impala* (proposed method) and *snesim* (relocating strategy). Figures 10 and 11 show the results obtained by using *impala* and *snesim*, respectively.

It is observed that the proposed method is stable if the number of multiple grid levels are changed: The occurrence proportion maps for each facies are quite similar when using 3 and 5 levels (Figs. 10d, 10e, 10f, and Figs. 10g, 10h, 10i). Hence, it is also the case for the standard deviation maps of the indicator of each facies (Figs. 10j, 10k, 10l, and Figs. 10m, 10n, 10o), since the variance of an indicator variable  $I$  is equal to the product  $p \cdot (1 - p)$ , where  $p$  is the probability  $p = P(I = 1)$

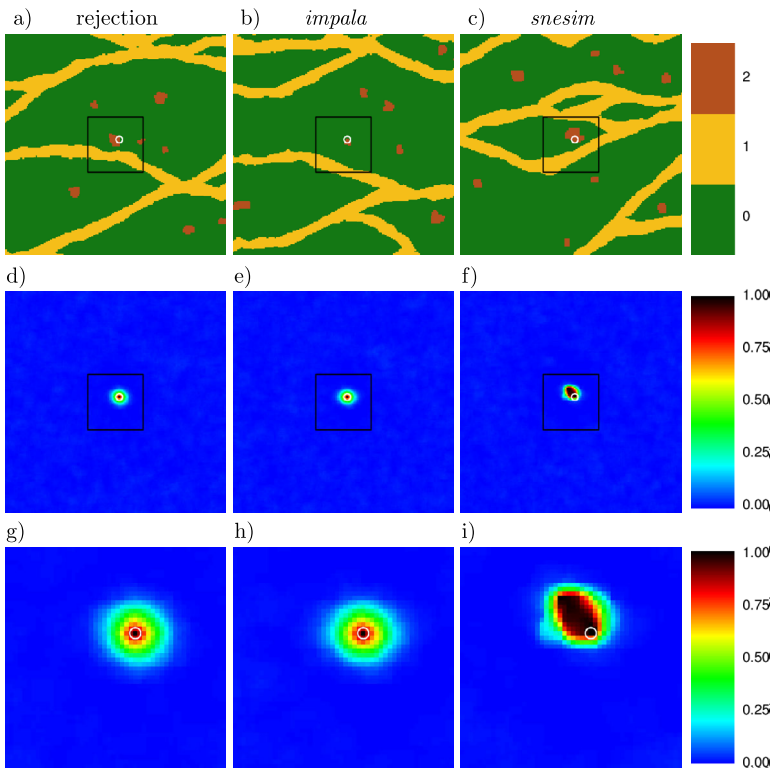


**Fig. 7** Results for the situation with one hard datum of value 2 (small structure) located as shown in the top row of Fig. 4. The node containing the hard datum is highlighted with a white circle in every map. (Top row) one realization ( $200 \times 200$ ) obtained by (a) rejection method, (b) *impala* and (c) *snesim*; (middle row) occurrence proportion map for facies 2 for 500 realizations generated by (d) rejection method, (e) *impala* and (f) *snesim*; (bottom row) (g–i) zoom on the squared area displayed in the previous row

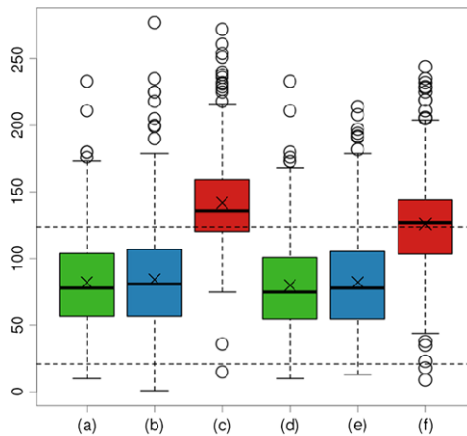
(and then, the standard deviation is deduced from the occurrence proportion). On the contrary, Fig. 11 shows that *snesim* is sensitive to the number of multiple grid levels. If we increase the number of levels, the variability in the simulations decreases; wider area around the hard data location are frozen (see the regions corresponding to low values in the standard deviation maps of Fig. 11). Finally, comparing standard deviation maps in Figs. 10 and 11, we observe that *impala* allows for more variability than *snesim*.

## 6 Conclusions

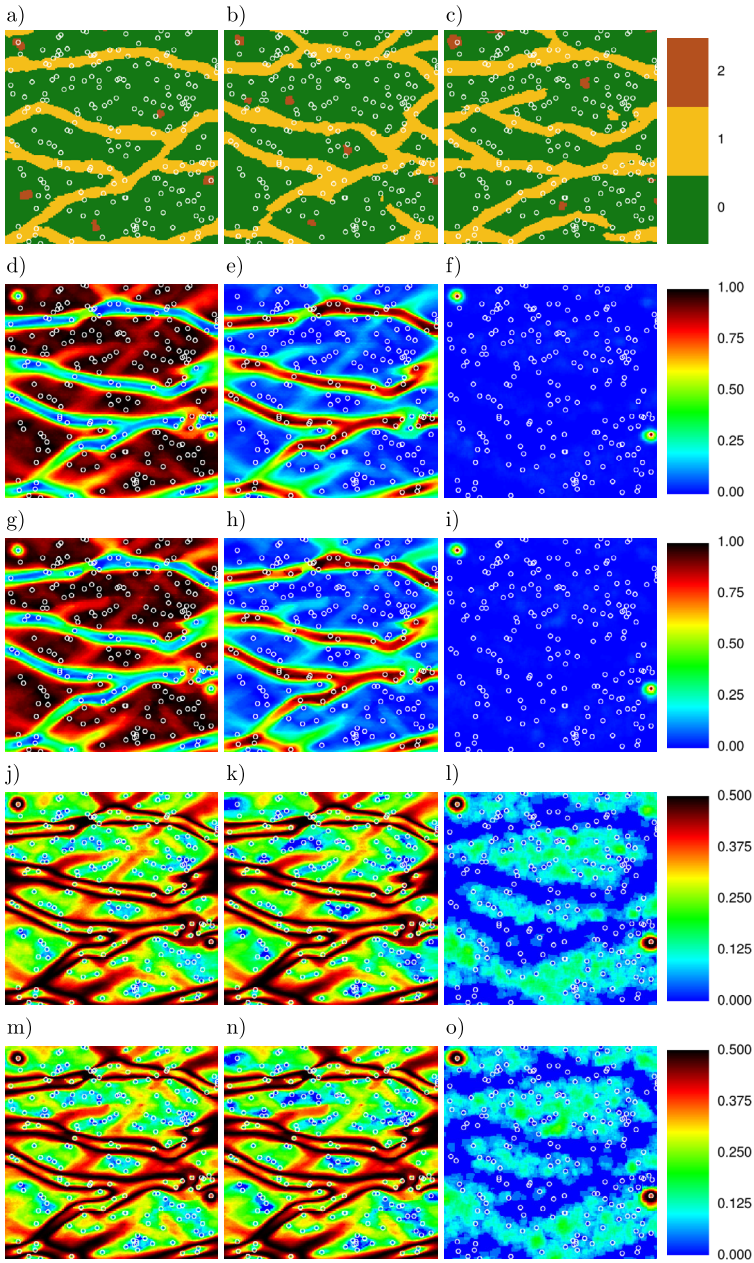
In this paper, we propose a method for addressing the hard data conditioning in MPS algorithms based on a multiple grid approach is proposed. This method consists in bringing the information of the hard data through all the multiple grid levels according to the conceptual model. First, the hard data are assigned to the nearest nodes



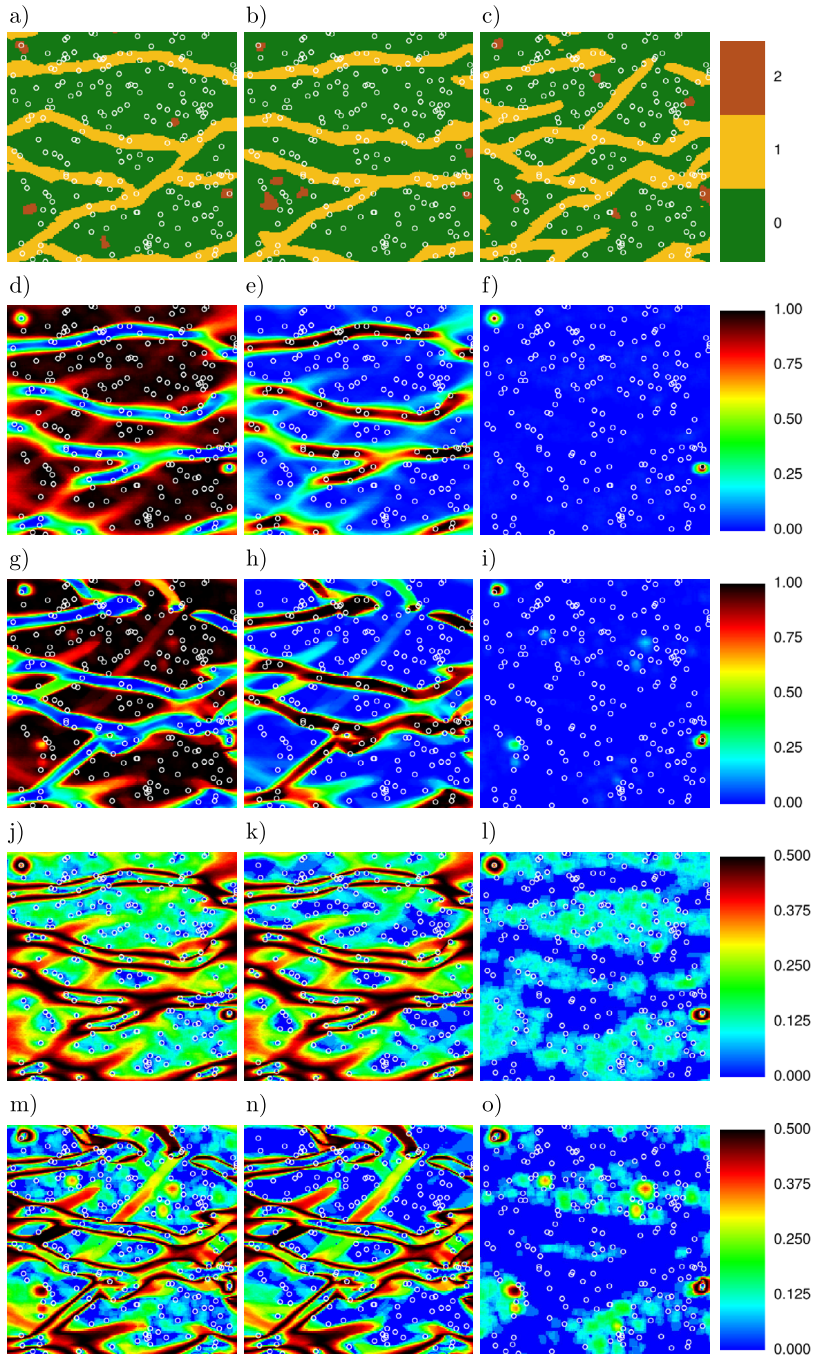
**Fig. 8** Results for the situation with one hard datum of value 2 (channel) located as shown in the bottom row of Fig. 4. The display is organized as in Fig. 7



**Fig. 9** Size of the geobody (facies 2) containing the hard datum for the 500 realizations obtained by (a) rejection method (left column of Fig. 7); (b) *impala* (middle column of Fig. 7); (c) *snesim* (right column of Fig. 7); (d) rejection method (left column of Fig. 8); (e) *impala* (middle column of Fig. 8); (f) *snesim* (right column of Fig. 8). Each box represent the inter–quartile of the geobody sizes in pixels, the line within the boxes their median, and the cross their mean. The two horizontal dashed lines represent the range (min and max) of the size of the geobodies for facies 2 found in the TI



**Fig. 10** Results of 500 realizations obtained by *impala* in a situation with 200 hard data in a simulation grid of dimensions  $200 \times 200$ , using the TI in Fig. 2. The hard data locations are highlighted by white circles in every map. (1st row) (a) reference image from which the hard data are extracted, (b–c) one realization with (b) 3 and (c) 5 multiple grid levels; (2nd row) (d–f) occurrence proportion maps with 3 multiple grid levels, for (d) facies 0, (e) facies 1, (f) facies 2; (3rd row) standard deviation maps with 3 multiple grid levels, for (g) facies 0, (h) facies 1, (i) facies 2; (4th row) standard deviation maps with 5 multiple grid levels, for the indicator of (j) facies 0, (k) facies 1, (l) facies 2; (5th row) standard deviation maps with 5 multiple grid levels, for the indicator of (m) facies 0, (n) facies 1, (o) facies 2



**Fig. 11** Results obtained by *snesim* in the same situation and with the same display as in Fig. 10



in the simulation grid, and then from the second fine level to the coarse level, the nodes enclosing the hard data are simulated using the MPS statistics of the level one rank finer. Then the simulation is completed by simulating the remaining uninformed nodes, from the coarse level to the fine level.

The method consisting, for each level from coarse to fine, in simply relocating the hard data to the nearest node before simulating the remaining uninformed nodes, leads to systematic biases depending on the location of the data, and to a decreasing variability around the data when the number of multiple grid levels is increasing. Unlike this latter method, we show that the proposed method is robust because: (1) the variability (uncertainty) of the simulations in the surroundings of the hard data is in accordance with the conceptual model and does not depend on the data location relative to the multiple grid levels, and (2) the method is not sensitive to the specified number of multiple grid levels. Hence, this technique, implemented in *impala*, deals with the hard data conditioning properly, guaranteeing a fair representation of the conceptual model.

**Acknowledgements** The authors would like to thank Philippe Renard for his helpful advice.

## References

- Alcolea A, Renard P (2010) Blocking moving window algorithm: conditioning multiple-point simulations to hydrogeological data. *Water Resour Res* 46:W08511. doi:[10.1029/2009WR007943](https://doi.org/10.1029/2009WR007943)
- Arpat GB, Caers J (2007) Conditional simulation with patterns. *Math Geol* 39(2):177–203. doi:[10.1007/s11004-006-9075-3](https://doi.org/10.1007/s11004-006-9075-3)
- Caers J (2003) History matching under training-image-based geological model constraints. *SPE J* 8(3):218–226
- Caers J, Hoffman T (2006) The probability perturbation method: a new look at Bayesian inverse modeling. *Math Geol* 38(1):81–100. doi:[10.1007/s11004-005-9005-9](https://doi.org/10.1007/s11004-005-9005-9)
- Cordua KS, Hansen TM, Mosegaard K (2012) Monte Carlo full-waveform inversion of crosshole gpr data using multiple-point geostatistical a priori information. *Geophysics* 77(2):H19–H31. doi:[10.1190/GEO2011-0170.1](https://doi.org/10.1190/GEO2011-0170.1)
- Guardiano F, Srivastava R (1993) Multivariate geostatistics: beyond bivariate moments. In: Soares A (ed) *Geostatistics Troia*, vol 1. Kluwer Academic, Dordrecht, pp 133–144
- Hu LY, Zhao Y, Liu Y, Scheepens C, Bouchard A (2013) Updating multipoint simulations using the ensemble Kalman filter. *Comput Geosci* 51:7–15. doi:[10.1016/j.cageo.2012.08.020](https://doi.org/10.1016/j.cageo.2012.08.020)
- Huysmans M, Dassargues A (2012) Modeling the effect of clay drapes on pumping test response in a cross-bedded aquifer using multiple-point geostatistics. *J Hydrol* 450:159–167. doi:[10.1016/j.jhydrol.2012.05.014](https://doi.org/10.1016/j.jhydrol.2012.05.014)
- Jafarpour B, Khodabakhshi M (2011) A probability conditioning method (PCM) for nonlinear flow data integration into multipoint statistical facies simulation. *Math Geosci* 43(2):133–164. doi:[10.1007/s11004-011-9316-y](https://doi.org/10.1007/s11004-011-9316-y)
- Lange K, Frydendall H, Cordua KS, Hansen TM, Melnikova Y, Mosegaard K (2012) A frequency matching method: solving inverse problems by use of geologically realistic prior information. *Math Geosci* 44(7):783–803. doi:[10.1007/s11004-012-9417-2](https://doi.org/10.1007/s11004-012-9417-2)
- Mariethoz G, Renard P, Caers J (2010a) Bayesian inverse problem and optimization with iterative spatial resampling. *Water Resour Res* 46:W11530. doi:[10.1029/2010WR009274](https://doi.org/10.1029/2010WR009274)
- Mariethoz G, Renard P, Straubhaar J (2010b) The direct sampling method to perform multiple-point geostatistical simulations. *Water Resour Res* 46:W11536. doi:[10.1029/2008WR007621](https://doi.org/10.1029/2008WR007621)
- Remy N, Boucher A, Wu J (2009) *Applied geostatistics with SGeMS. A user's guide*. Cambridge University Press, Cambridge
- Renard P, Straubhaar J, Caers J, Mariethoz G (2011) Conditioning facies simulations with connectivity data. *Math Geosci* 43(8):879–903. doi:[10.1007/s11004-011-9363-4](https://doi.org/10.1007/s11004-011-9363-4)

- Ronayne MJ, Gorelick SM, Caers J (2008) Identifying discrete geologic structures that produce anomalous hydraulic response: an inverse modeling approach. *Water Resour Res* 44(8):W08426. doi:[10.1029/2007WR006635](https://doi.org/10.1029/2007WR006635)
- Saripally I, Caers J (2008) Evaluating data conditioning accuracy of MPS algorithms and the impact on flow modeling. Paper presented at 21th SCRF meeting, Stanford University, May 8–9, 2008
- Straubhaar J, Renard P, Mariethoz G, Froidevaux R, Besson O (2011) An improved parallel multiple-point algorithm using a list approach. *Math Geosci* 43(3):305–328. doi:[10.1007/s11004-011-9328-7](https://doi.org/10.1007/s11004-011-9328-7)
- Straubhaar J, Walgenwitz A, Renard P (2013) Parallel multiple-point statistics algorithm based on list and tree structures. *Math Geosci* 45(2):131–147. doi:[10.1007/s11004-012-9437-y](https://doi.org/10.1007/s11004-012-9437-y)
- Strebelle S (2002) Conditional simulation of complex geological structures using multiple-points statistics. *Math Geol* 34(1):1–21. doi:[10.1023/A:1014009426274](https://doi.org/10.1023/A:1014009426274)
- Tahmasebi T, Hezarkhani A, Sahimi M (2012) Multiple-point geostatistical modeling based on the cross-correlation functions. *Comput Geosci* 16(3):779–797. doi:[10.1007/s10596-012-9287-1](https://doi.org/10.1007/s10596-012-9287-1)
- Tran TT (1994) Improving variogram reproduction on dense simulation grids. *Comput Geosci* 20(7–8):1161–1168. doi:[10.1016/0098-3004\(94\)90069-8](https://doi.org/10.1016/0098-3004(94)90069-8)
- Wu J, Zhang T, Journel A (2008) Fast filtersim simulation with score-based distance. *Math Geosci* 40(7):773–788. doi:[10.1007/s11004-008-9157-5](https://doi.org/10.1007/s11004-008-9157-5)
- Zhang T, Switzer P, Journel AG (2006) Filter-based classification of training image patterns for spatial simulation. *Math Geol* 38(1):63–80. doi:[10.1007/s11004-005-9004-x](https://doi.org/10.1007/s11004-005-9004-x)
- Zhou H, Gomez-Hernandez JJ, Franssen H-JH, Li L (2011) An approach to handling non-gaussianity of parameters and state variables in ensemble Kalman filtering. *Adv Water Resour* 34(7):844–864. doi:[10.1016/j.advwatres.2011.04.014](https://doi.org/10.1016/j.advwatres.2011.04.014)