Using Multiple Grids in Markov Mesh Facies Modeling

Odd Kolbjørnsen · Marita Stien · Heidi Kjøsberg ·
Bjørn Fjellvoll · Petter Abrahamsen

Received: 9 January 2013 / Accepted: 20 October 2013 / Published online: 9 November 2013
© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract A multigrid Markov mesh model for geological facies is formulated by defining a hierarchy of nested grids and defining a Markov mesh model for each of these grids. The facies probabilities in the Markov mesh models are formulated as generalized linear models that combine functions of the grid values in a sequential neighborhood. The parameters in the generalized linear model for each grid are estimated from the training image. During simulation, the coarse patterns are first laid out, and by simulating increasingly finer grids we are able to recreate patterns at different scales. The method is applied to several test cases and results are compared to the training image and the results of a commercially available snesim algorithm. In each test case, simulation results are compared qualitatively by visual inspection, and quantitatively by using volume fractions, and an upscaled permeability tensor. When compared to the training image, the method produces results that only have a few percent deviation from the values of the training image. When compared with the snesim algorithm the results in general have the same quality. The largest computational cost in the multigrid Markov mesh is the estimation of model parameters from the training image. This is of comparable CPU time to that of creating one snesim realization. The simulation of one realization is typically ten times faster than the estimation.

O. Kolbjørnsen · M. Stien · H. Kjøsberg · B. Fjellvoll · P. Abrahamsen (✉)
Norwegian Computing Center, PO Box 114, Blindern, 0314 Oslo, Norway
e-mail: Petter.Abrahamsen@nr.no

O. Kolbjørnsen
e-mail: Odd.Kolbjornsen@nr.no

M. Stien
e-mail: Marita.Stien@nr.no

H. Kjøsberg
e-mail: Heidi.Kjonsberg@nr.no

B. Fjellvoll
e-mail: Bjorn.Fjellvoll@nr.no

Keywords Multiple grids · Markov mesh models · Sequential simulation · Generalized linear models

1 Introduction

The spatial distribution of facies is a crucial part of any reservoir model since it is often one of the main sources of variability in flow (Skorstad et al. 2005). Multipoint statistics is one class of methods for geological facies modeling, proposed nearly two decades ago (Guardiano and Srivastava 1993), and it has developed along two main paths: the statistical model approach (Tjelmeland and Besag 1998) and the algorithmic approach (Strebelle and Journel 2000). Common to many multipoint methods is the use of a training image that represents the geologic patterns typically found in the reservoir of study. The multipoint methods aim at reproducing the essential aspects of these patterns, but with a variability that can be adapted to the case at hand. During simulation, algorithmic multipoint methods create situations where the conditioning event is not found in the training image. This creates artifacts in the simulations (Strebelle and Remy 2005). Statistical models on the other hand, can interpolate between observed patterns to compute the probability of patterns that are not explicitly present in the training image, and hence artifacts can potentially be reduced. With the introduction of the Markov mesh model (Stien and Kolbjørnsen 2011), the statistical model based approach also overcame its original time-consumption problems in parameter estimation and simulation. In this paper, we proceed yet another step, and formulate a multigrid Markov mesh model.

The use of multiple grids has previously been used in the algorithmic approach to multipoint methods (Strebelle 2002), and for general geostatistical approaches (Tran 1994). The strategy has proved invaluable for capturing patterns at different scales. With a multigrid Markov mesh model, we combine an advantage developed for algorithmic methods—the use of multiple grids, with the consistency and flexibility of the statistical model. Combining multiple grids with Markov random fields was recently explored by Toftaker and Tjelmeland (2013). Markov mesh models are a subclass of Markov random fields (Tjelmeland and Besag 1998) defined through a unilateral path (Daly 2005; Daly and Knudby 2007). In Stien and Kolbjørnsen (2011), the authors propose to model facies geometries through a single-grid Markov mesh model defined using the framework of generalized linear models (McCullagh and Nelder 1989; Cressie and Davidson 1998). In this paper, a hierarchy of grids is defined, and a Markov mesh model analogous to that of Stien and Kolbjørnsen (2011) is defined for each grid, but such that it takes into account information also from coarser grids. The result is what we denote a multigrid Markov mesh model. The framework of generalized linear models and systematic grid specification enable fast parameter estimation. The estimation is done once per grid level. During simulation, the coarse patterns are first laid out, and by simulating increasingly finer grids we are able to create patterns at different scales. We present several three-dimensional examples, illustrating that the multigrid Markov mesh model can be successfully applied for a range of training images. For each training image, the simulation results are quantitatively evaluated by comparing facies fractions and up-scaled permeability tensors of realizations and training images.

2 Multigrid Markov Mesh Model

Multigrid Markov mesh models are defined by a hierarchy of grids, a unilateral path per grid level, and a conditional probability for each cell value given the cell values in a sequential neighborhood. Mathematically, the multigrid Markov mesh model is nothing but a single-grid Markov mesh model where the cells are visited according to the overall path, and the sequential neighborhood for any cell consists only of cells from the past part of this path. It is nevertheless useful to explicitly discuss the model in terms of the multiple grid levels, since it is the systematic model specification in terms of these levels that makes it an efficient and useful tool for capturing patterns at different scales.

2.1 The Single-Grid Model

Single-grid Markov mesh models are defined by a unilateral path and a conditional probability for each cell value given the cell values in a sequential neighborhood. Consider a finite, regular grid $\mathcal{G}$ in two or more dimensions and let the one-dimensional index i label the cells of the grid. The set of all cells is $\{1, 2, \dots, N\}$. For the single-grid Markov mesh models, this is also the order in which cells are visited during simulation. For each cell i , we let the cell value x_i represent the facies of the cell. Assuming that the conditional probability for facies at cell i depends only on a subset Γ_i of all cells $j < i$, we can write this probability as

$$\pi(x_i | x_{j < i}) = \pi(x_i | x_{\Gamma_i}), \quad (1)$$

where x_{Γ_i} is the set of facies values for the cells in the sequential neighborhood. Equation (1) expresses the Markov property of the model. Figure 1 gives an illustration of a sequential neighborhood on a two-dimensional grid. The single-grid Markov mesh model is fully specified through the conditional probabilities

$$\pi(x_1, x_2, \dots, x_N) = \prod_{i=1}^N \pi(x_i | x_{\Gamma_i}). \quad (2)$$

Simulation from the Markov mesh model is performed by following the path $i = 1, 2, \dots, N$ throughout the grid. For each cell, the facies value is drawn according to

Fig. 1 Illustration of sequential neighborhood. A snapshot of a simulation is displayed, and the gray cells have not yet been simulated

Fig. 2 Left: a sequential neighborhood. Right: the same sequential neighborhood, but on a coarser grid

the conditional probability $\pi(x_i|x_{I_i})$. Each cell is visited once, and the resulting grid configuration follows the joint probability distribution in Eq. (2).

2.2 Capturing Large Scale Patterns

The main purpose of using a multigrid formulation is to detect and reproduce patterns at a large scale with a relatively small neighborhood. Figure 2 illustrates how this works for a simple two-dimensional example. The left figure pane shows a sequential neighborhood I_i of a cell i on the grid $\mathcal{G}$. The right figure pane shows the analogous neighborhood, but applied to the coarser grid level l . The grid $\mathcal{G}_l$ consists of every 2nd and every 4th cell of $\mathcal{G}$, in the horizontal and vertical direction, respectively. Relative to the grids $\mathcal{G}$ and $\mathcal{G}_l$, respectively, the neighborhoods of the left and right figure pane are identical. However, relative to the cells on the finest grid, $\mathcal{G}$, the neighborhoods on $\mathcal{G}_l$ reach much farther. It is a significantly faster to use a model with a small neighborhood at each grid level than to use a single-grid model with a single large neighborhood. So, by using multiple grids, we can capture patterns on large scales while retaining a model that is computationally efficient.

2.3 Defining the Multigrid Markov Mesh Model

We are now ready to present the formulation of the multigrid Markov mesh model. Consider again a finite, regular grid $\mathcal{G}$ in two or more dimensions, with the one-dimensional index i labeling the cells of the grid. The set of all cells is $\{1, 2, \dots, N\}$. We define a sequence of regular grids $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_L$ where each grid is a subset of cells in $\mathcal{G}$ such that

$$\mathcal{G}_1 \subset \mathcal{G}_2 \subset \mathcal{G}_3 \cdots \subset \mathcal{G}_L, \quad \text{and} \quad \mathcal{G}_L = \mathcal{G}. \quad (3)$$

We will refer to $\mathcal{G}_l$ as the grid on level l , with the coarsest grid level being for $l = 1$ and the finest for $l = L$. Define furthermore the disjoint sets $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_L$ by

$$\mathcal{H}_1 = \mathcal{G}_1, \quad \text{and} \quad \mathcal{H}_l = \mathcal{G}_l \setminus \mathcal{G}_{l-1}, \quad \text{for } l = 2, 3, \dots, L. \quad (4)$$

The set $\mathcal{H}_l$ consists of the cells of $\mathcal{G}_l$ that are not on any of the coarser grid levels. Also note that $\mathcal{G}_l = \bigcup_{k=1}^L \mathcal{H}_k$. Figure 3 illustrates a two-dimensional example of how a grid can be refined. We first select a coarsest level and then refine one direction

Fig. 3 A hierarchy of grids, displayed as subsets of the set of cells of the finest grid $\mathcal{G}$

1	2	3	4	5	6	7	8	9	1	46	10	47	2	48	11	49	3
10	11	12	13	14	15	16	17	18	26	50	27	51	28	52	29	53	30
19	20	21	22	23	24	25	26	27	16	54	17	55	18	56	19	57	20
28	29	30	31	32	33	34	35	36	31	58	32	59	33	60	34	61	35
37	38	39	40	41	42	43	44	45	4	62	12	63	5	64	13	65	6
46	47	48	49	50	51	52	53	54	36	66	37	67	38	68	39	69	40
55	56	57	58	59	60	61	62	63	21	70	22	71	23	72	24	73	25
64	65	66	67	68	69	70	71	72	41	74	42	75	43	76	44	77	45
73	74	75	76	77	78	79	80	81	7	78	14	79	8	80	15	81	9

Fig. 4 *Left*: one-dimensional cell index. *Right*: path order of cells, given the grid sequence of Fig. 3

at the time until we reach the finest grid level. Details on how the grid refinement directions are determined is described in Sect. 3.4.

For each set $\mathcal{H}_l$, let $s(l)$ denote the number of cells in $\mathcal{H}_l$, and let p_l be the ordered version of the set of cells on $\mathcal{H}_l$

$$p_l = (i_1, i_2, \dots, i_{s(l)}), \quad \text{where} \\ i_1, i_2, \dots, i_{s(l)} \in \mathcal{H}_l, \text{ and } i_q < i_r \text{ if } q < r \quad \forall q \in \{1, 2, \dots, s(l)\}. \quad (5)$$

Here, i_q is the original one-dimensional cell index. The last condition says the cells on $\mathcal{H}_l$ are ordered with increasing one-dimensional index. But be aware that p_l consists only of cells that are on $\mathcal{H}_l$. The ordered set p_l is the unilateral path for grid level l . The total path of the multigrid Markov mesh model is defined to be the concatenation of the level-wise paths

$$p = (p_1, p_2, \dots, p_L). \quad (6)$$

The right-hand side of Fig. 4 shows the order of visiting the cells in a two-dimensional grid, given that the one-dimensional cell indices are as shown in the left-most figure pane, and the grid sequence is as in the example of Fig. 3.

Since the sets $\mathcal{H}_l$ are disjoint and their union equals $\mathcal{G}$, for each grid cell i , there exists a unique grid level l such that $i \in \mathcal{H}_l$ and $i \notin \mathcal{H}_k$ if $k \neq l$. We write $l(i)$ for this

mapping from cell index to grid level, that is $i \in \mathcal{H}_{l(i)}$. Given that we follow the path p , all cells that are listed before cell i are given by the set

$$\mathcal{W}_i = \{k | k \in \mathcal{G}_{l(i)-1} \cup \mathcal{P}_i\}_{k=1}^N, \quad (7)$$

where

$$\mathcal{P}_i = \{j | j \in \mathcal{H}_{l(i)}, j < i\}_{j=1}^N. \quad (8)$$

That is, any cell listed before cell i either belongs to a coarser grid, or is listed before cell i in the unilateral path $p_{l(i)}$ for the grid level associated with i . For each cell i , we let the cell value x_i represent the facies of the cell. The joint probability distribution can always be written as

$$\pi(x_1, x_2, \dots, x_N) = \prod_{l=1}^L \prod_{i \in p_l} \pi(x_i | x_{\mathcal{W}_i}), \quad (9)$$

since this only amounts to a reordering of the grid cells on the left-hand side of the expression, followed by repeated use of the general relation $\pi(A, B) = \pi(A|B)\pi(B)$. Equation (9) shows how the joint probability distribution of all cells can be expressed in terms of a systematic grid refinement, where the conditional probability $\pi(x_i | x_{\mathcal{W}_i})$ depends only on cells from the present or coarser grid levels, never on cells from finer grid levels. Now we introduce the Markov condition: We assume that the conditional probability for facies at cell i depends only on a certain subset of the cells from earlier in the path, that is we assume that there exists a $\Gamma_i \subset \mathcal{W}_i$ such that

$$\pi(x_i | x_{\mathcal{W}_i}) = \pi(x_i | x_{\Gamma_i}). \quad (10)$$

Then the joint probability can be written

$$\pi(x_1, x_2, \dots, x_N) = \prod_{l=1}^L \prod_{i \in p_l} \pi(x_i | x_{\Gamma_i}). \quad (11)$$

The set Γ_i is denoted the sequential neighborhood of cell i . It consists of cells from the coarser grids and cells that are listed before cell i on the path p_l . Equation (11) is a multigrid Markov mesh model. It is fully specified by the sequence of grid levels and the conditional probabilities $\pi(x_i | x_{\Gamma_i})$. Simulation from the model is carried out by starting at the coarsest grid level, then using increasingly finer grids. At each grid level, the unilateral path p_l is followed, and for each cell the facies value is drawn according to the conditional probability $\pi(x_i | x_{\Gamma_i})$. Each cell is visited once, and the resulting grid configuration follows the joint probability distribution in Eq. (11).

2.4 Data Conditioning

Since our methodology requires a systematic scan of the grid, the issue of data conditioning of irregular patterns arise. This issue is not unique to the multigrid Markov mesh approach, but arise also when using multiple grids in standard multipoint algorithms. The common solution to this problem for snesim type of algorithms is to

relocate the hard data to the nearest simulation grid node (Strebelle 2002). An alternative solution for the data conditioning that is applicable to many multiple grid models was proposed by Kjnsberg and Kolbjrnsen (2008). The general expression to be evaluated is the probability of a specific facies in cell i , conditioned to previously simulated cells $\mathbf{x}_{\mathcal{W}_i}$ and well data ahead in the path, $\mathbf{x}_{d_i}$. Following Kjnsberg and Kolbjrnsen (2008), we write the posterior probability as

$$p(x_i | \mathbf{x}_{\mathcal{W}_i}, \mathbf{x}_{d_i}) = \frac{p(x_i | \mathbf{x}_{\mathcal{W}_i}, \mathbf{x}_{d_i})}{p(x_i | \mathbf{x}_{\mathcal{W}_i})} p(x_i | \mathbf{x}_{\mathcal{W}_i}), \quad (12)$$

where the right-hand side is a trivial rewriting of the left-hand side. The posterior probability is the product of a marginal likelihood for the remaining well data (leftmost factor) and the unconditional probability (rightmost factor). The unconditional probability is obtained from the multigrid Markov mesh model as described in Eq. (10), whereas the probabilities in the numerator and denominator in the leftmost factor is approximated by predictions obtained from indicator Kriging. The resulting expression becomes

$$p(x_i | \mathbf{x}_{\mathcal{W}_i}, \mathbf{x}_{d_i}) \propto \frac{Z(x_i | \mathbf{x}_{\mathcal{W}_i}, \mathbf{x}_{d_i})}{Z(x_i | \mathbf{x}_{\mathcal{W}_i})} p(x_i | \mathbf{x}_{\Gamma_i}), \quad (13)$$

where $Z(x_i | \mathbf{x}_{\mathcal{W}_i})$ is the predictor for x_i given by indicator Kriging conditioned on cells already simulated and $Z(x_i | \mathbf{x}_{\mathcal{W}_i}, \mathbf{x}_{d_i})$ is the Kriging predictor conditioned also on future data points. The proportional sign in the relation indicates that the result must be renormalized due to the approximation introduced by the indicator Kriging predictions. The methodology can be used to condition on both soft and hard data. A statistical correct data conditioning can be achieved by integrating the approximation in a Metropolis–Hastings algorithm according to Kjnsberg and Kolbjrnsen (2008).

3 Model Specification

The statistical model is defined by specifying the path and parameterization of the conditional probabilities in Eq. (10). The path is uniquely determined by the sequence of grid. The statistical model specification is based on generalized linear models (McCullagh and Nelder 1989). The formulation is such that the parameters are efficient to estimate and simple to interpret.

3.1 Specifying the Sequential Neighborhood

The multigrid sequential neighborhood is a generalization of the sequential neighborhood of the single-grid Markov mesh formulation, the latter being illustrated on the left-hand side of Fig. 2. The generalization consists of including in the neighborhood cells with a higher one-dimensional index, provided they belong to the past path of the reference cell. We illustrate this with an example in Fig. 5, where the one-dimensional indices and path of Fig. 4 are reused. The figure gives an example of the sequential neighborhood Γ_{42} . In the example, the sequential neighborhood has

1	2	3	4	5	6	7	8	9	1	46	10	47	2	48	11	49	3
10	11	12	13	14	15	16	17	18	26	50	27	51	28	52	29	53	30
19	20	21	22	23	24	25	26	27	16	54	17	55	18	56	19	57	20
28	29	30	31	32	33	34	35	36	31	58	32	59	33	60	34	61	35
37	38	39	40	41	42	43	44	45	4	62	12	63	5	64	13	65	6
46	47	48	49	50	51	52	53	54	36	66	37	67	38	68	39	69	40
55	56	57	58	59	60	61	62	63	21	70	22	71	23	72	24	73	25
64	65	66	67	68	69	70	71	72	41	74	42	75	43	76	44	77	45
73	74	75	76	77	78	79	80	81	7	78	14	79	8	80	15	81	9

Fig. 5 Example of sequential neighborhood Γ_{42} . *Left pane*: one-dimensional cell indices; *Right pane*: order of cells in the path

a maximal extension of two cells in each direction from the reference cell. Now, cell 42 is ordered as number 64 in the path, as indicated in the right-hand figure pane. The sequential neighborhood includes only cells with a lower path order number, but as shown by the left-hand figure pane, some of the cells have a one-dimensional index larger than 42: cells 43, 50, 52, 59, and 61 in this example.

For each grid level, we specify the maximal extension in different directions, and then include all cells inside these limits provided they belong to the past path. Since each grid $\mathcal{G}$ is a regular grid, assuming the one-dimensional indexing is also regular, all cells $i \in \mathcal{H}_i$ are then assigned sequential neighborhoods of exactly the same shape. This is important for the efficiency of the model. When all cells have the same neighborhood, we only have to estimate one model at each grid level. This is faster than estimating several models and the estimates are more precise since we have more available data. For a sequential neighborhood bounded by a rectangular box, six parameters are sufficient for parameterization, the extent of the box and the location of the reference cell. If the box is assumed symmetric around the reference cell, three parameters describing the extent of the box (l_x, l_y, l_z) establishes the neighborhood.

Figure 6 illustrates symmetrically bounded sequential neighborhood in three-dimensional by displaying four z -layers for a given grid $\mathcal{G}_i$. Only cells on grid $\mathcal{G}_i$ are shown so only cells already simulated or cells simulated at the current level are displayed. In this example $l_x = l_y = 3$ and $l_z = 2$, and the one-dimensional cell index loops first over the x -direction, then the y -direction, and last the z -direction. In this situation, three different types of a sequential neighborhood may occur depending on whether the refinement from grid $\mathcal{G}_{i-1}$ to grid $\mathcal{G}_i$ is done in direction z , y , or x . In Case 1, it is assumed that the refinement in the z -direction. In mathematical terms, this means that if $\mathcal{G}_{i-1}$ consists of each $2^{k_x}, 2^{k_y}, 2^{k_z}$ cells of the finest grid $\mathcal{G}$, then $\mathcal{G}_i$ consists of each $2^{k_x}, 2^{k_y}, 2^{k_z-1}$ cells. In Case 2, the refinement was done in the y -direction, and in Case 3 the refinement was done in the x -direction. Figure 6 gives the three generic alternative neighborhoods, for refinement from grid $\mathcal{G}_{i-1}$. Only one of these is selected for the model building. See Sect. 3.4 for further discussion.

3.2 Using the Framework of Generalized Linear Models

The use of a generalized linear model (GLM) for specifying Markov mesh models for facies modeling was first suggested in Stien and Kolbjørnsen (2011). The idea in

Fig. 6 The sequential neighborhood for each of three alternative grid refinement cases, provided the neighborhood's maximal extension is $l_x = l_y = 3$ and $l_z = 2$. The sequential neighborhood consists of the cells in the past path

GLM is that the distribution of a response variable depends on a linear combination of explanatory variables through a nonlinear link function. We let the facies x_i be the response variable, and the explanatory variables be functions of the sequential neighborhood Γ_i . Consider a given grid level l and let the cell i be on this grid level that is $i \in \mathcal{H}_l$. Let $\mathbf{z}_i$ be a P_l -dimensional vector of explanatory variables with elements that are functions of cells from the sequential neighborhood Γ_i . We propose particular functions below, but for now simply write $z_{ij} = f_j(x_{\Gamma_i})$ for $j \in \{1, 2, \dots, P_l\}$. The same set of P_l functions is used for any neighborhood Γ_i if $i \in \mathcal{H}_l$. The value $f_j(x_{\Gamma_i})$ varies with i , since it depends on the facies configuration in the neighborhood Γ_i . In the model, we propose there is one model parameter for every pair of neighborhood function and facies value at each grid level l . Assuming there are K different facies values, we let the K vectors $\theta_l^1, \dots, \theta_l^K$ hold the parameters. Each vector is P_l -dimensional.

We encode the cell value $x_i \in \{1, 2, \dots, K\}$ with binary variables x_i^k such that

$$x_i^k = \begin{cases} 1 & \text{if } x_i = k, \\ 0 & \text{else.} \end{cases} \quad (14)$$

The conditional probability in Eq. (10) is then

$$\pi(x_i | \Gamma_i) = \pi(x_i | \mathbf{z}_i, \theta_{l(i)}^1, \dots, \theta_{l(i)}^K) = \frac{\prod_{k=1}^K \exp\{x_i^{k_1} \mathbf{z}_i^T \theta_{l(i)}^{k_1}\}}{\sum_{k_2=1}^K \exp\{\mathbf{z}_i^T \theta_{l(i)}^{k_2}\}}, \quad (15)$$

and the joint probability in Eq. (11) is

$$\pi(x_1, x_2, \dots, x_N) = \prod_{l=1}^L \prod_{i \in p_l} \frac{\prod_{k=1}^K \exp\{x_i^{k_1} \mathbf{z}_i^T \theta_l^{k_1}\}}{\sum_{k_2=1}^K \exp\{\mathbf{z}_i^T \theta_l^{k_2}\}}. \quad (16)$$

Here, p_l is the path on grid level l . Equation (16) is a GLM for each grid level. The maximum likelihood estimation of the parameters in the multigrid Markov mesh formulation is solved with the iterative weighted least squares scheme. This is the same approach which is used in Stien and Kolbjørnsen (2011). The way our model is formulated makes it possible to estimate the model independently at each grid level. The only change which is made in comparison to Stien and Kolbjørnsen (2011) is the selected neighborhood functions.

3.3 Neighborhood Functions

As noticed by Stien and Kolbjørnsen (2011), the challenge with multipoint statistics is that there generally are too many possible patterns. A finite training image does not hold information about all possibilities. To overcome this problem, we extract a subset of properties, represented by neighborhood functions that are important in order to reproduce geological structures and aim at making robust choices. The specification of the neighborhood functions is similar to the specification of Stien and Kolbjørnsen (2011) in the sense that the three-dimensional model consists of combining three two-dimensional models. For each of three orthogonal grid slices, intersecting at the reference cell, we add off-two-dimensional extensions and consider two point interactions, multipoint interactions representing continuity and transitions of facies, and multipoint interactions representing all possible patterns for a very limited number of cells.

3.3.1 Two-Dimensional Specification

For each two-dimensional grid slice of the sequential neighborhood of the reference cell i , the two-dimensional two-point interactions are restricted to a subset of the slice in question. For each cell j of this subset, indicator functions $f^k(x_j)$ are included, one for each facies $k \in \{1, 2, \dots, K\}$. The functions are defined by

$$f^k(x_j) = \begin{cases} 1 & \text{if } x_j = k, \\ 0 & \text{else.} \end{cases} \quad (17)$$

This gives one function per facies per two-point interaction cell, with one model parameter per function. For each two-dimensional grid slice, the multipoint functions representing connectivity and transition of facies are also represented by indicator functions. Each function is defined relative to a certain set of cells. Let γ_i^{l-1} be a set of $l-1$ neighbors of the reference cell i , corresponding to an l -point interaction term (Stien and Kolbjørnsen 2011). The indicator function $f^k(x_{\gamma_i^{l-1}})$ is then defined by

$$f^k(x_{\gamma_i^{l-1}}) = \begin{cases} 1 & \text{if } x_j = k, \forall j \in \gamma_i^{l-1}, \\ 0 & \text{else.} \end{cases} \quad (18)$$

The sets γ_i^{l-1} are defined by considering strips of cells in the horizontal, vertical, and diagonal directions of the two-dimensional slice (Stien and Kolbjørnsen 2011). When the size of the sequential neighborhood increases, more functions are included and

the number of parameters to be estimated increase. Indicator functions are also used for the multipoint interactions representing all possible patterns for a very limited number of cells in the two-dimensional grid slice. There is one function per facies per pattern of the neighboring cells, resulting in K^n functions if $n - 1$ neighbors participate to this kind of multipoint interaction.

3.3.2 Three-Dimensional Specification

The three-dimensional model is defined by using the combined set of indicator functions from the three two-dimensional grid slices. We also include strips in the off-two-dimensional diagonal directions of each of the eight octants centered around the reference cell i , using only the octant cells that are part of the multigrid sequential neighborhood. This is analogous to the specification of Stien and Kolbjørnsen (2011). The resulting number of functions for the three-dimensional model is denoted P_l , where l labels the grid level. The set of neighborhood functions generally differs across the grid levels, but for a given level l the same set of P_l functions is used for all cells $i \in \mathcal{H}$. Figure 7 illustrates some of the two-dimensional neighborhood functions for the case of a grid $\mathcal{G}_l$ that is assumed related to grid $\mathcal{G}_{l-1}$ by a refinement in the y -direction (second row of Fig. 6). Only the z -plane of the reference cell is shown in Fig. 7. The left column of the figure illustrates functions that are used when the sequential neighborhood is identical to its symmetrical bounding box; the rightmost column illustrates what the actual neighborhood functions look like after the form of the sequential neighborhood for this grid level is taken into account, and the middle column illustrates the filtering that takes us from the general functions of the left column to the grid level specific functions of the right. The arrows indicate the directions and in which order the number of interaction terms increases. The cells marked by the x means that this cell is included as a 2-particle interaction with the center cell, and for the bottom-most row as an n -particle interaction with all the other cells marked by x and the center cell.

3.4 Grid Refinement

The purpose of the multigrid formulation is to be able to reproduce patterns on large scales using small neighborhoods or templates. This is done by initially applying the neighborhood or template on a coarse grid and applying the same neighborhood or template on successively finer grids. The grids are refined by inserting grid levels to increase the grid resolution (Fig. 3). There is no unique way of refining the grid that will work equally well for all training images. Two different grid refinement sequences will represent two different models. We will proceed by selecting one refinement scheme that is adapted to the training image and estimate the parameters for this model. The alternative of using multiple refinement schemes, each with an attached probability, is abandoned in order to have a parsimonious model representation. The challenge is therefore to find a nice way of obtaining grid $\mathcal{G}_l$ from the grid at the previous level, $\mathcal{G}_{l-1}$ by a refinement in the x -, y -, and z -directions. It is possible to refine in a single direction or to combine two or three directions at the same time. The simplest is either to choose a single refinement in one direction or

Fig. 7 A two-dimensional example of how general pattern recognition functions are modified by the sequential neighborhood to form the grid specific neighborhood functions

single refinement in all three directions. Treating all directions equal is possible, but training images are seldom isotropic so we have chosen to refine in one direction at a time. This also gives the most flexible model and the additional computational cost is acceptable since most of the CPU time is spent on the final grid resolutions. Refining in one direction means that grid $\mathcal{G}_l$ has twice as many grid cells as the grid at the previous level, $\mathcal{G}_{l-1}$. We start by selecting a coarse grid and choose the direction of refinement along the direction of the weakest spatial correlation. This has the consequence that the direction of refinement will change several times so that spatial structures in all directions interact at several resolutions. This consequence is not obvious and is best explained by considering the opposite approach: Assume that all directional correlations are monotonically decreasing with distance. If the direction of maximum correlation is selected, then the direction that have the maximum correlation at the coarsest grid level will be refined until the finest grid level before any of the two other directions are refined. So, this approach will prevent structures in different directions from interacting. The second important property is that by selecting the direction of the weakest spatial correlation we keep more randomness after more

steps in the refinement sequence. This will hopefully reduce the chance of getting stuck into improbable patterns at an early stage of the simulation.

The algorithm used for selecting the grid refinement schemes starts with a grid level, (g_x, g_y, g_z) , where g_k is the number of times the grid has been coarsened in the k -direction. So, choosing for instance grid level $(3, 3, 2)$ means that the grid must be refined $3 + 3 + 2 = 8$ times to obtain the original fine resolution grid. This also means that the fine resolution grid has $2^3 \cdot 2^3 \cdot 2^2 = 256$ times more grid cells than the starting grid. The starting grid level must be chosen manually as we have not been able to find a robust way to estimate starting values for an arbitrary training image. The selection of grid refinement as used in our approach is summarized in the following algorithm:

1. Estimate the three directional correlation functions, $c_x(h_x)$, $c_y(h_y)$, and $c_z(h_z)$, in the x -, y -, or z -directions from the training image. The lags, h_x , h_y , and h_z , are measured as distances in terms of the finest grid resolution in each direction.
2. Compute the absolute value of these functions: $g_k(h) = |c_k(h)|$, for $k \in \{x, y, z\}$.
3. Compute monotonized correlation functions: $r_k(h) = \max_{u \geq h} g_k(u)$, for $k \in \{x, y, z\}$.
4. For grid level (g_x, g_y, g_z) to grid level $(0, 0, 0)$:
 - Get refinement direction $k_s = \arg \min_{k \in \{x, y, z\}} r_k(2^{g_k} - 1)$.
 - Set $g_{k_s} = g_{k_s} - 1$.

The monotonization in the second and third step is a practical solution to avoid problems with oscillating correlation functions. The algorithm is illustrated in Fig. 8 for a two-dimensional example. The top figure shows the absolute value of the estimated correlation functions, whereas the bottom function shows the monotonized function. For the selection, we have two proposal lags, one for x -direction and one for y -direction, and we see that by choosing to refine in the x -direction first we get a slightly lower increase in correlation than by refining in the y -direction.

4 Examples

This section presents results obtained by using the multigrid Markov mesh model. We consider four different three-dimensional training images, three with two facies classes, and one with three facies classes. The training images are displayed in Figs. 9 to 12. We provide a comparison between the results of the multigrid Markov mesh model and the snesim approach (Strebelle and Journel 2000 and Strebelle 2002). The latter is obtained by running the multipoint module of the RMS modeling software (<http://www2.emersonprocess.com/en-US/brands/roxar/>). Settings for the generation of realizations are summarized in Table 1 that also provides the dimensions of the training image and the simulation grid. Note that the settings found in Table 1 is somewhat arbitrary. We have chosen parameters so that both approaches give good results using acceptable amounts of memory and CPU time. Note in particular that we needed more grid refinements for the snesim approach than the Markov mesh approach to obtain acceptable simulated realizations in these four cases. For each training image, we have generated 100 realizations with the multigrid Markov mesh

Fig. 8 Directional correlation functions of a two-dimensional training image. This is used to determine how far the neighborhood must reach in order to capture large scale patterns

Table 1 Listing of the test cases, and settings for the simulation runs. Column 1 is the simulation name of the test case. Column 2 is the dimensions of the training image and simulation grid (which are identical). Column 3 gives the maximum neighborhood extension for the Markov mesh model. Column 5 gives the number of cells in the template used by the snesim algorithm. Column 4 and 6 gives the number of grid refinements in the x - y - and z -direction for the Markov mesh model and the snesim algorithm. The number of grid levels where chosen to obtain similar results

Test case	Grid size	Markov mesh		snesim	
		Neighborh. size (l_x, l_y, l_z)	Grid levels (g_x, g_y, g_z)	Template size	Grid levels (g_x, g_y, g_z)
Turbidite	$100 \times 100 \times 100$	(3, 3, 3)	(2, 2, 2)	102	(3, 3, 2)
Azimuth Channels	$100 \times 100 \times 100$	(2, 2, 2)	(2, 2, 2)	102	(6, 6, 4)
Isolated Channels	$125 \times 125 \times 75$	(4, 2, 2)	(5, 5, 4)	100	(6, 6, 6)
Channel Crevasse	$100 \times 100 \times 100$	(4, 4, 2)	(2, 2, 2)	80	(6, 5, 4)

model, and 100 realizations with the snesim algorithm. We compare the results qualitatively and quantitatively by comparing the single cell upscaled permeability and the facies volume fractions. There are no conditioning data so all simulations are unconditional in these tests.

4.1 Conceptual Geology

Conceptual geology is assessed by visual inspection and comparison between the training image and the realizations. Figures 9, 10, 11, and 12 compare the conceptual geology of multigrid Markov mesh realizations, snesim realizations, and the training

Fig. 9 Conceptual geology, turbidite case. Training image: *left*, multigrid Markov mesh model: *middle*, snesim algorithm: *right*

Fig. 10 Conceptual geology, azimuth channels case. Training image: *left*, multigrid Markov mesh model: *middle*, snesim algorithm: *right*

images for each of the test cases. Each of the four figures show the training image to the left, a realization obtained by the multigrid Markov mesh model in the middle, and a realization obtained by the snesim algorithm to the right. The training images represent, respectively, a turbidite system (Fig. 9), a channels system where the main correlations are along the x , y -diagonal, and where the channels are somewhat narrow and irregular (Fig. 10), a system of irregular channels with crevasses (Fig. 11), and a system of regular channels with fairly low sinuosity (Fig. 12). The realization displayed for the proposed approach and the snesim approach are representative for the set 100 realizations obtained. The simulated realizations clearly illustrate that the multigrid Markov mesh model reproduces main aspects of the training image, such as correlation directions and body shapes. There is no major difference between the visual appearance of the multigrid Markov mesh realizations and the realizations obtained by the multigrid snesim approach.

4.2 Upscaled Permeability

The purpose of generating facies realizations is to be able to model the large contrasts in the permeability field in an efficient way. The difference in high and low permeability facies can be several orders of magnitude. The most important factor for modeling of flow response is the geometric continuity of the most permeable facies. The single cell upscaled permeability tensor gives a measure of the direc-

Fig. 11 Conceptual geology, channel crevasse case. Training image: *left*, multigrid Markov mesh model: *middle*, snesim algorithm: *right*

Fig. 12 Conceptual geology, isolated channels case. Training image: *left*, multigrid Markov mesh model: *middle*, snesim algorithm: *right*

tional flow properties of the realizations, and is sensitive to the continuity of objects in the specified flow direction. To compute the single cell upscaled permeability, we generate the full scale permeability field by populating the facies realizations with a homogeneous permeability for each facies class; see Table 2 for values. The computations are performed using single phase assumptions, periodic boundary conditions, and executed using the open source software “Opm-upscaling,” http://www.opm-project.org/wiki/index.php/Upscaling_single_phase_permeability. The resulting tensors are symmetric, 3-dimensional tensors of second order. In the comparison, we only include the diagonal elements, representing permeability in the xx -, yy -, and zz -direction. The off-diagonal elements provide no further insight, and are hence omitted. The upscaled permeability is calculated also for the training image for comparison. Note that a direct comparison requires that the dimensions of the training image and the simulated realizations are identical.

Figures 13 and 14 display the histogram for the upscaled permeability tensors obtained from the 100 realizations together with the similar response from the training image. For each training image, the results for the multigrid Markov mesh model are displayed in the top row, and the results for the snesim algorithm are displayed in the bottom row. For each tensor component, the probability distributions obtained from simulations are shown as blue histograms, and the red line is the corresponding permeability of the training image. All cases are able to recreate the large difference between horizontal and vertical flow. This reflects the horizontal continuity in the

Table 2 The permeability values in milli Darcy used for each facies in the test cases when computing the upscaled permeability

Test case	Sand	Shale	Crevasse
Turbidite	1000 mD	1 mD	–
Azimuth Channels	1000 mD	1 mD	–
Isolated Channels	1000 mD	1 mD	–
Channel Crevasse	1000 mD	1 mD	700 mD

training images. For the turbidite case in particular (left pane Fig. 13) both the multigrid Markov mesh model and the snesim algorithm reproduce the training image's permeability tensor well; the red line is well inside the distributions, often corresponding to the mean of each distribution. There is a small exception for the snesim zz -permeability, but the percent-wise deviation to the training image permeability is almost negligible. For the azimuth channels case, the distributions for the xx - and yy -permeability contain the training image permeability for both the multigrid Markov mesh model and the snesim algorithm. But for both methods the training image permeability is in the tail of the distributions. The multigrid Markov mesh model tends to make realizations with too little connectivity, the snesim approach tends to make too much connectivity. It is reasonable that the results for the xx -permeability is similar to the results for the yy -permeability, since the training image in this case has its main correlations along the x , y -diagonal, and hence is symmetric for x and y . The training image permeability in the zz -direction is under-estimated/over-estimated by the multigrid Markov mesh model and the snesim algorithm. For the channel crevasse case and the isolated channels, both the multigrid Markov mesh model and the snesim algorithm capture the scale of the permeabilities, with small percent-wise deviations from the training image permeabilities. For these training images, the two methods behave very similar: They both have a tendency to underestimate the xx - and yy -permeabilities for the channel crevasse case and for the yy -permeability of the isolated channels; they both overestimate the xx -permeability of the isolated channels; and they both tend to slightly overestimate the zz -permeabilities.

4.3 Volume Fractions

For each realization, we also compute the facies volume fractions, and compare this to the volume fractions of the training image. Statistical distributions for upscaled permeability and volume fractions is obtained separately for the multigrid Markov mesh model and for the snesim approach, one distribution per training image. Table 3 displays the mean and standard deviation of the sand volume fraction in the realizations. Both multigrid Markov mesh model and snesim are all very close to the target and the variability is small for both methods, and smallest for the multigrid Markov mesh.

4.4 Time Usage

Figure 15 show histograms of the time it took to estimate and simulate the first realization and subsequent realizations. Generating subsequent realizations is faster for

Fig. 13 Upscaled permeability. *Left:* turbidite case, *right:* azimuth channels case. *Upper row:* multigrid Markov mesh model (MMM); *lower row:* snesim algorithm (MPS). *Red line:* training image's upscaled permeability

Fig. 14 Upscaled permeability. *Left:* channels crevasse case, *right:* isolated channels case. *Upper row:* multigrad Markov mesh model (MMM); *lower row:* snesim algorithm (MPS). *Red line:* training image's upscaled permeability

Table 3 Comparison of sand fraction in training image av realizations

Test case	Training image	Realizations	Markov mesh	Realizations snesim	
	Mean	Mean	Std. dev.	Mean	Std. dev.
Turbidite	0.1679	0.1591	0.0004	0.1764	0.0024
Azimuth Channels	0.1470	0.1358	0.0007	0.1558	0.0017
Isolated Channels	0.2693	0.2566	0.0022	0.2723	0.0041
Channel Crevasse	0.3757	0.3799	0.0001	0.3812	0.0009

Fig. 15 Time usage in minutes for the four cases

the multigrid Markov mesh model once the model is estimated. The isolated channels case is a special case where the channels are very regularly shaped. This gives little variation in patterns so that the search tree build by snesim is small. For this case, the snesim algorithm becomes very efficient. This comparison is done using the snesim implementation available in Irap RMS. For this implementation, the building of the search tree takes approximately 25 % of the CPU time. Other implementations may exploit this and there are possibly more efficient implementations available. One should also note that there has been a long development of various (commercial) snesim implementations whereas the Markov mesh model still is an immature approach.

5 Conclusions

We have presented a multigrid Markov mesh model for geological facies modeling. This combines an advantage originally developed for algorithmic multipoint methods—the use of multiple grids—with the flexibility and consistency of the statistical approach to multipoint methods. The model consists of a hierarchy of nested grids, with a single-grid Markov mesh model for each grid, but such that information from coarser grids are taken into account. We have adopted the specification of earlier published single-grid Markov mesh models, by using the framework of generalized linear models and a parameterization that captures continuity/discontinuity of geological structures. The result is a model that gives results that are comparable to the established snesim algorithm for several test cases.

Acknowledgements We would like to thank Statoil and Roxar for financial support, and for making available the tool for doing permeability upscaling and the Irap RMS software. We would also like to thank the Norwegian Research Council for financial support.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Cressie N, Davidson J (1998) Image analysis with partially ordered Markov models. *Comput Stat Data Anal* 29:1–26
- Daly C (2005) Higher order models using entropy, Markov random fields and sequential simulation. In: Leuangthong O, Deutsch C (eds) *Quantitative geology and geostatistics, geostatistics Banff 2004*. Springer, Dordrecht, pp 215–224
- Daly C, Knudby C (2007) Multipoint statistics in reservoir modelling and in computer vision. In: *EAGE petroleum geostatistic*. Cascais, Portugal, 10–14 September 2007, A32
- Guardiano F, Srivastava RM (1993) Multivariate geostatistics: beyond bivariate moments. In: Soares A (ed) *Geostatistics-Troia*, vol 1. Kluwer, Dordrecht, pp 133–144
- Kjærneberg H, Kolbjørnsen O (2008) Markov mesh simulations with data conditioning through indicator kriging. In: *Proceedings of the eighth international geostatistics congress*, vol 1. Kluwer, Dordrecht, pp 257–266
- McCullagh P, Nelder J (1989) *Generalized linear models*, 2nd edn. Chapman and Hall, London
- Skorstad A, Kolbjørnsen O, Fjellvoll B, Howell J (2005) Sensitivity of oil production to petrophysical heterogeneities. In: Leuangthong O, Deutsch C (eds) *Quantitative geology and geostatistics, geostatistics Banff 2004*. Springer, Dordrecht, pp 723–730
- Stien M, Kolbjørnsen O (2011) Facies modeling using a Markov mesh model specification. *Math Geosci* 43:611–624
- Strebel S (2002) Conditional simulation of complex geological structures using multiple-point statistics. *Math Geol* 34:1–22
- Strebel S, Journé A (2000) Sequential simulation drawing structures from training images. In: Kleingeld W, Krige D (eds) *Geostatistics Capetown*, vol 1, pp 381–392
- Strebel S, Remy N (2005) Post-processing of multiple-point geostatistical models to improve reproduction of training patterns. In: Leuangthong O, Deutsch C (eds) *Quantitative geology and geostatistics, geostatistics Banff 2004*. Springer, Dordrecht, pp 723–730
- Tjelmeland H, Besag JE (1998) Markov random fields with higher-order interactions. *Scand J Stat* 25:415–433
- Toftaker H, Tjelmeland H (2013) Construction of binary multi-grid Markov random field prior models from training images. *Math Geosci* 45:383–409
- Tran TT (1994) Improving variogram reproduction on dense simulation grids. *Comput Geosci* 20:1161–1168