ORIGINAL PAPER

Multiscale ensemble filtering for reservoir engineering applications

**Wiktoria Lawniczak · Remus Hanea ·
Arnold Heemink · Dennis McLaughlin**

Abstract Reservoir management requires periodic updates of the simulation models using the production data available over time. Traditionally, validation of reservoir models with production data is done using a history matching process. Uncertainties in the data, as well as in the model, lead to a nonunique history matching inverse problem. It has been shown that the ensemble Kalman filter (EnKF) is an adequate method for predicting the dynamics of the reservoir. The EnKF is a sequential Monte-Carlo approach that uses an ensemble of reservoir models. For realistic, large-scale applications, the ensemble size needs to be kept small due to computational inefficiency. Consequently, the error space is not well covered (poor cross-correlation matrix approximations) and the updated parameter field becomes scattered and loses important geological features (for example, the contact between high- and low-permeability values). The prior geological knowledge present in the initial time is not found anymore in the final updated parameter. We propose a new approach to overcome some of the EnKF limitations. This paper shows the specifications and results of the ensemble multiscale filter (EnMSF) for automatic history matching. EnMSF replaces, at each update time, the prior sample covariance with a multiscale tree. The global dependence is preserved via the parent–child relation in the tree (nodes at the adjacent scales). After constructing the tree, the Kalman update is performed.

The properties of the EnMSF are presented here with a 2D, two-phase (oil and water) small twin experiment, and the results are compared to the EnKF. The advantages of using EnMSF are localization in space and scale, adaptability to prior information, and efficiency in case many measurements are available. These advantages make the EnMSF a practical tool for many data assimilation problems.

Keywords Ensemble · Multiscale · Kalman

1 Introduction

History matching (HM) is a process of adjusting the variables in a reservoir simulation model until it closely reproduces the past behavior of the reservoir. The accuracy of the history matching depends on the quality of the reservoir model and the quality and quantity of the data available. In traditional history matching, the production data for the entire history are matched at the same time, and repeated simulations are needed, which makes the process time-consuming. There are gradient-based HM methods that require a minimization of a cost function. In a real and large-scale application, it is an expensive procedure, and it can be stuck in local minima. Due to the presence of uncertainties in both the data and the model, it is hard and expensive to use traditional methods that might involve repeated HM exercises with different initial models.

One way to solve these problems is to use sequential data assimilation schemes, in particular, the ensemble Kalman filter (EnKF). In the past few years, successful applications of Kalman filter theory were reported in many areas of research: the meteorological applications

W. Lawniczak (✉) · R. Hanea · A. Heemink
TU Delft, Delft, the Netherlands
e-mail: wiktoria.lawniczak@gmail.com

D. McLaughlin
M.I.T., Cambridge, MA, USA

[2, 9], nonlinear shallow-water storm-surge models [25], and atmospheric chemistry and transport modeling (e.g., [12, 23, 26]).

The EnKF has also entered the world of reservoir engineering. Several publications have discussed the use of EnKF with oil reservoir models: Nævdal et al. [19–21], Gu and Oliver [10], Gao and Reynolds [8], Liu and Oliver [15], Wen and Chen [27], and Skjervheim et al. [24], showing promising results and, at the same time, raising some possible drawbacks.

The EnKF is based on the representation of the probability density of the state estimate by a finite number N (N being much smaller than the number of elements in the state vector) of randomly generated system states (ensemble members). This method falls into the Bayesian inversion approach and may provide a solution to the combined parameter and state estimation problem. The result is an ensemble of analyzed solutions (the combination between the measurements and the reservoir model), which best approximates the posterior probability density function for the model parameters.

The ensemble size limits the number of degrees of freedom used to represent forecast and analysis errors. It makes the calculation of the error covariances practical for modest-sized ensembles. One important consequence of the use of small-sized ensembles is the sampling error problem. After a certain number of assimilation steps, the ensemble loses its variance and leads to filter divergence. Houtekamer and Mitchell [13] conclude that the use of a small number of members in an ensemble often produces spuriously large magnitude background error covariances between greatly separated grid points (unphysical correlations). They noted that the EnKF analysis scheme could be improved by excluding observations at great distances from the grid point being analyzed by performing a "covariance localization." Examples of this approach include methods based on covariance filtering with Schur products [11, 14] and methods that perform updates in small blocks of grid cells [17, 22]. These methods improve computational efficiency and suppress the negative effect of sampling errors. The covariances that are used for localization will have an impact on the description of the physical correlation carried by the forecast covariance. Therefore, there is a risk of introducing correlations that are not physically possible. A number of researchers have observed and discussed the imbalances introduced by the localization schemes in the meteorological applications [16, 18].

In this paper, we focus on data assimilation with the ensemble multiscale filter (EnMSF) [29] for estimation of the oil reservoir parameters (permeability). This new approach solves some of the limitations of EnKF discussed above by allowing for a spatial localization that preserves the correct correlations.

Multiscale estimation is based on the concept of using a multiscale tree that describes the spatial correlations. The method is based on an algorithm [28] inspired by image processing research. The degree of freedom to choose a certain tree and to set up the parameters for the update of the ensemble makes the method very appealing. At the same time, one should be aware of the strong dependence of the performance of the method on the choices mentioned above.

In this paper, we will show an interesting feature of the algorithm and the influence of its setting on the quality of the estimates in case of a reservoir engineering application. Due to the complexity of the method, we look only at a one time step update of the ensemble. In Section 2, the theoretical background for the EnKF and for the EnMSF and the assumptions that need to be made are presented. In Section 3, a 2D, two-phase example is presented with seismic data. The numbering of the cells in the numerical grid is discussed. The conclusions follow in Section 4.

2 EnKF and EnMSF—theoretical background

2.1 Kalman filter for nonlinear systems

Kalman filtering represents a link between a model and measurements. First, a state vector x needs to be defined (a collection of variables representing the model). The following superscripts for x will be used in the equations: f represents the forecast state and a represents the analysis state. Kalman filtering processes the measurements in a physically consistent way taking into account the model dynamics. This is achieved by extending the deterministic model represented by

$$x_{k+1} = \mathcal{M}(x_k, k)$$

to a stochastic model

$$x_{k+1} = \mathcal{M}(x_k, k) + w_k,$$

where $\mathcal{M}$ represents the nonlinear model, which propagates the state of the system from time k to $k + 1$, and w_k is the white-noise process, $w_k \sim N(0, Q_k)$, which quantifies the uncertainties in the model. The covariance matrix Q_k needs to be specified.

All the available data for time k are stored in a vector y_k. In general, there are fewer observations than variables in the model. The only correct way to compare the values measured with the state vector is to use a

function from the model space to observation space called the observation operator H:

$$y_k = H_k x_k. \tag{1}$$

Through the observation operator H_k, a forecast for the observed data locations can be made from the forecast of the state. Uncertainties in the measurements need to be specified as well. Therefore, the vector y_k from Eq. 1 is expanded as follows:

$$y_k = H_k x_k + v_k.$$

The observation operator H_k is a collection of interpolation operators from the model discretization to the observation points (conversions from model variables to the observed parameters); v_k is the observation noise process, $v_k \sim N(0, R_k)$. The covariance matrix R_k needs to be specified.

2.2 Ensemble Kalman filter

The EnKF was introduced by Evensen [4] and has been successfully used in many applications [6, 13]. This Monte Carlo approach is based on the representation of the probability density of the state estimate by an ensemble of possible states, $\xi_1, \xi_2, \ldots, \xi_N$. Each ensemble member is assumed to be a single sample from a distribution of the true state. Whenever necessary, statistical moments are approximated with sample statistics. We can rewrite the steps of the Kalman filter algorithm for the EnKF, as shown below.

- Initialization step: an ensemble of N states is generated to represent the uncertainty in x_0,

$$\xi_{1,0}^a, \xi_{2,0}^a, \ldots, \xi_{N,0}^a.$$

- Forecast step:

$$\xi_{i,k}^f = M\left(\xi_{i,k-1}^a\right) + w_{i,k},$$
$$x_k^f = \frac{1}{N} \sum_{i=1}^N \xi_{i,k}^f,$$
$$E_k^f = \left[\xi_{1,k}^f - x_k^f, \xi_{2,k}^f - x_k^f, \ldots, \xi_{N,k}^f - x_k^f\right],$$
$$P_k^f = \frac{1}{N-1} E_k^f \left(E_k^f\right)^\top,$$
$$i = 1, \ldots, N. \tag{2}$$

- Analysis step: when the measurements become available, the mean and the covariance are replaced with the equivalent ones in the analysis step:

$$K_k = P_k^f H_k^\top \left[H_k P_k^f H_k^\top + R_k\right]^{-1},$$
$$\xi_{i,k}^a = \xi_{i,k}^f + K_k \left[y_k - H_k \xi_{i,k}^f + v_{i,k}\right].$$

Here, $w_{i,k}$ and $v_{i,k}$ are the realizations of the noise processes w_k and v_k, respectively. Based on the new updated values $\xi_{i,k}^a$, the analysis covariance matrix P_k^a is built.

The advantages of this algorithm are that P_k^f and P_k^a are always positive semidefinite and that the linear tangent model is not required because the ensembles are propagated using the original model, as in Eq. 2. In the final implementation of the algorithm, P_k^f values do not need to be computed [4]. For most practical problems, the forecast equation (Eq. 2) is computationally dominant [3]. As a result, the computational effort required for the EnKF is approximately N model simulations.

2.3 Ensemble multiscale filter

The EnMSF [29] provides an alternative way to perform the update step. The original ensemble covariance is represented by a tree structure, and physically long distance dependencies are kept through the relations between the tree nodes.

It consists of three basic steps:

1. Assigning grid cells (pixels) to the finest scale nodes and computing the tree parameters from sample propagation through the tree (tree construction)
2. Upward sweep (moving information upwards in the tree)
3. Downward sweep (spreading information downwards in the tree to the finest scale)

The ensemble members are partitioned with respect to grid geometry and settings (like the pixel numbering and the tree specification). The multiscale algorithm places the partition at the finest scale nodes (leaf nodes) and computes the parameters at the upper tree nodes. Now, the upward and downward steps can be performed and the output is a set of updated replicates.

2.3.1 Example

Since the ensemble multiscale algorithm is more complex than EnKF, first, a little example is shown. The

Fig. 1 The initial grid division

Fig. 3 The middle scale representation on the grid

example greatly simplifies the method but allows us to grasp the general idea.

Each pair of figures, Figs. 1–2, 3–4, and 5–6, shows a grid and corresponding tree states. This is the first stage of the EnMSF—tree construction. The 4×4 grid is a representation of permeability, where grey is high permeability and black is low.

The grid cells (pixels) are numbered and each group of four is assigned to a leaf node of a binary tree (Figs. 1 and 2). Here, it should be noted that having an ensemble of size N:

$$x_1^1 \quad x_1^2 \quad \quad x_1^N$$
$$x_2^1 \quad x_2^2 \quad \quad x_2^N$$
$$\vdots \quad , \quad \vdots \quad , \cdots , \quad \vdots \quad ,$$
$$x_{16}^1 \quad x_{16}^2 \quad \quad x_{16}^N$$

the first leaf node, for example, contains a matrix with the first four states of each ensemble member:

$$x_1^1 \quad x_1^2 \quad \quad x_1^N$$
$$x_2^1 \quad x_2^2 \quad \quad x_2^N$$
$$x_3^1 , \quad x_3^2 , \quad \cdots , \quad x_3^N .$$
$$x_4^1 \quad x_4^2 \quad \quad x_4^N$$

The state at each higher scale node is a linear combination of states at its direct children. At the middle scale, the four most influential states are kept at each
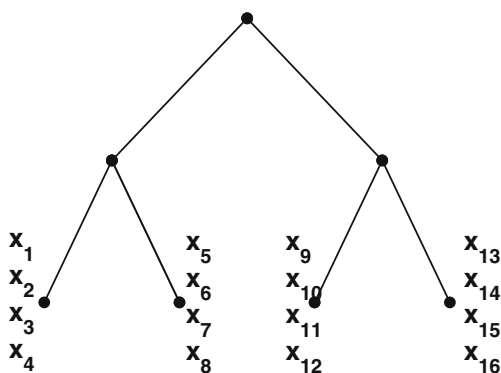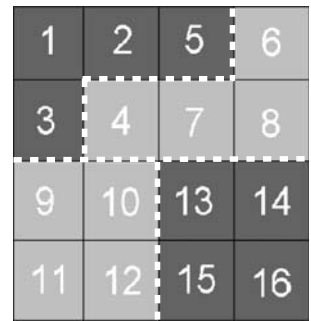
of the two nodes (Figs. 3 and 4). They happen to be the high-permeability channel. These eight values are used to compute the four states at the root node (Figs. 5 and 6), which is the center of the high-permeability channel. This is the end of the tree construction part when all the nodes contain sets of parameters needed to perform the upward and downward sweeps.

Assume that a measurement is available in pixel 1. It is placed at the node that had pixel 1 assigned to it, the first leaf node (a circle in Fig. 7). Going up the tree, a Kalman-based update is performed, and at the end, the root node contains the knowledge from the measurement. Downward sweep (Fig. 8) spreads the knowledge from the root node to all the other nodes. In consequence, the finest scale contains the analyzed states x_{i^a} (x_i^a), $i = 1, 2, ..., 16$.

Clearly, the ensemble filter operates on an ensemble representing a distribution of the truth. For simplicity, the example shows one grid representation. It should be clear though that the states at the tree nodes come from the dependencies in the ensemble.

2.3.2 The end of the example and some mathematics in the algorithm

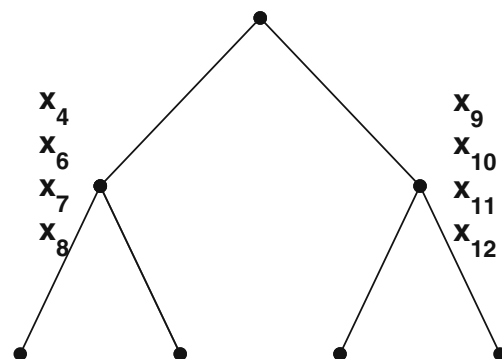The most complex is step 1, containing crucial assumptions and many flexible variables. Steps 2 and 3 are

Fig. 2 Tree with pixel values assigned to the leaf nodes (fines scale nodes)

Fig. 4 Pixels selected for the middle scale of the tree

Fig. 5 Root node representation on the grid

based on Kalman filter theory. Some mathematical details are presented here to enrich the simple example shown above. The full detailed description can be found in [29]. Some necessary notation is shown in Fig. 9.

Additionally, some symbols used in the text are:

Notation

$\chi(s)$	State vector at node s
$\chi_M(s)$	The vector of finest-scale states descended from s
$\chi(s\vert s)$	The state at node s after the upward sweep
$\chi(s\vert S)$	The state at node s after the downward sweep
$\chi(s\gamma\vert s)$	Projected state at node $s\gamma$
j	Superscript indicating an ensemble

Any other symbols are explained in the text.

The whole process starts with assigning the grid cells (pixels) to the leaf nodes of the tree. The cells can be numbered in various ways that determine the assignment. Two choices are shown in the next section. Assigned pixels provide states at the fine scale nodes of the tree.
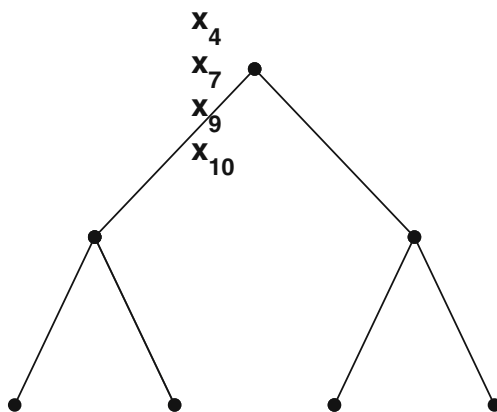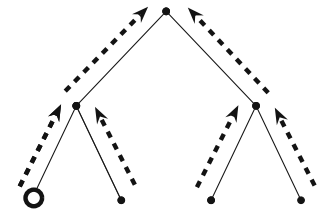
Fig. 7 A scheme of the upward sweep

A state at each non-fine-scale node s is a linear combination of the states at its children:

$$\chi(s) = V(s) \begin{bmatrix} \chi(s\alpha_1) \\ \vdots \\ \chi(s\alpha_q) \end{bmatrix},$$

where matrices $V(s)$ are obtained based on a process described in Appendix A. The size of $\chi(s)$ is controlled by the setup of $V(s)$, that is, it was $V(s)$ that allowed keeping four states at the upper scale nodes in the example. When all the states are computed and the measurements are placed at the tree nodes, the upward and downward sweeps can be carried out.

Going up the tree, the algorithm updates the states at the nodes. Then, each node s gets the value $\chi^j(s\vert s)$. $\chi^j(s\vert s)$ is the state vector updated with all the measurements in the subtree rooted at s. At the top of the tree, the value for the root node is obtained, $\chi^j(0\vert 0)$. This is the basis to perform the downward sweep of the algorithm. $\chi^j(0\vert 0)$ is the initial point, namely, $\chi^j(0\vert S)$. Going down the tree, the value $\chi^j(s\vert S)$ is assigned to each node s. That is the state value containing the knowledge from all given measurements. This way, at the end of the sweep, we get updated ensemble states at the finest scale, which can be used to perform the next forecast step.
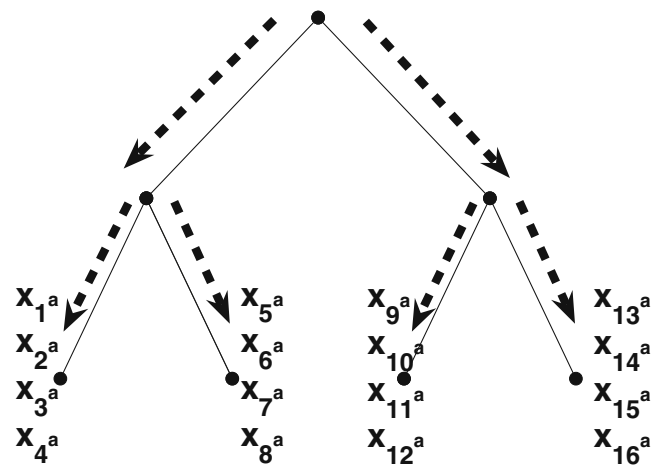

Fig. 6 Pixels selected for the root (*top*) node of the tree

Fig. 8 A scheme of the downward sweep and final updated values
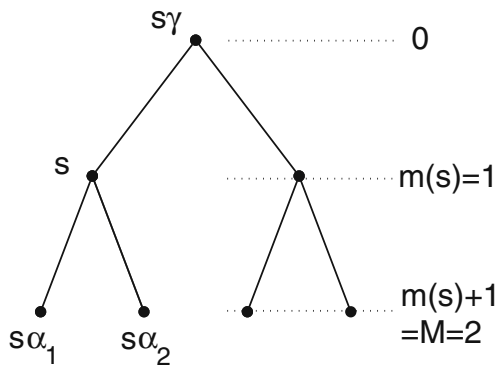
Fig. 9 Notation: $s\alpha_i$, the ith child of node s; $s\gamma$, the parent of node s; $m(s)$, the scale where s is placed; M, finest scale; 0, the root node

The equations leading the upward and downward sweeps are:

The upward sweep equation

$$\chi^j(s|s) = \chi^j(s) + K(s)\left[Y^j(s) - \hat{Y}^j(s)\right]$$

The states $\chi^j(s)$ at each node s are updated with perturbed measurements $Y^j(s)$ using weighting factor $K(s)$ and predicted measurements $\hat{Y}^j(s)$ (Appendix B).

The downward sweep equation

$$\chi^j(s|S) = \chi^j(s|s) + J(s)\left[\chi^j(s\gamma|S) - \chi^j(s\gamma|s)\right]$$

Previous states $\chi^j(s|s)$ at each node obtain the knowledge from all measurements through the weighting parameter $J(s)$ (Appendix C). $\chi^j(0|S) = \chi^j(0|0)$ is initially known from the upward sweep and
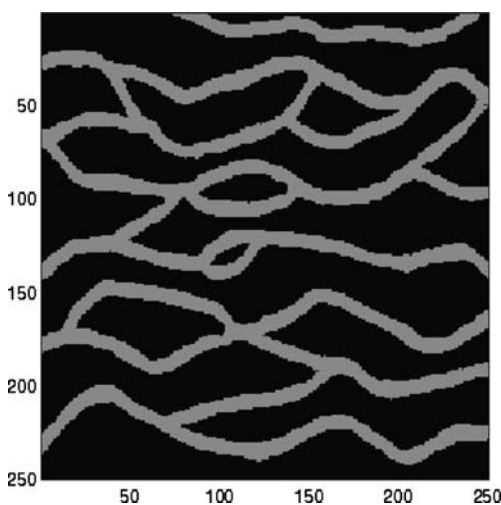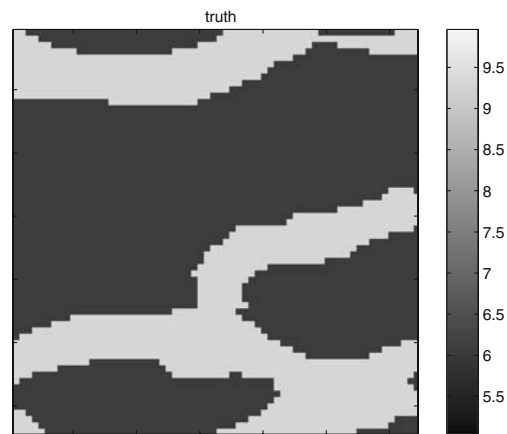

Fig. 10 The training image 250×250

Fig. 11 The truth 64×64

projected replicates $\chi^j(s\gamma|s)$ can be computed based on matrices $V(s)$ (Appendix D).

The whole procedure explained above, with the three steps, is able to approximate the forecast error covariance by constructing the tree and then to get the updated ensemble by moving up and down the tree assimilating the available measurements. In the end, the updated ensemble is obtained at the finest scale and the corresponding analyzed covariance matrix can be calculated.

3 Application

A twin experiment is prepared for the algorithm to check its performance. The results are compared to the EnKF's, as it is described in [5]. All shown results are one update time results.

Given the training image[1] (Fig. 10), an ensemble was generated using The Stanford Geostatistical Modeling Software. Algorithm *snesim* [1] generated 2D samples of permeability fields with grid size 64×64 and from the training image with grid size 250×250. Each of 100 replicates is built of two values of permeability: high 10,000 mD (grey color) and low 500 mD (black). The first replicate was assumed to be the "truth" (Fig. 11) and removed from the ensemble.

The values of the observations are the perturbed values of the "truth." It means that the permeability field is updated with permeability measurements. In practice, these values cannot be measured. Therefore, this example is not realistic but allows us to test almost any possible setup. Throughout the tests, the tree is a

---

[1]Training image is an image representing the features and the distribution of ensemble members [1].
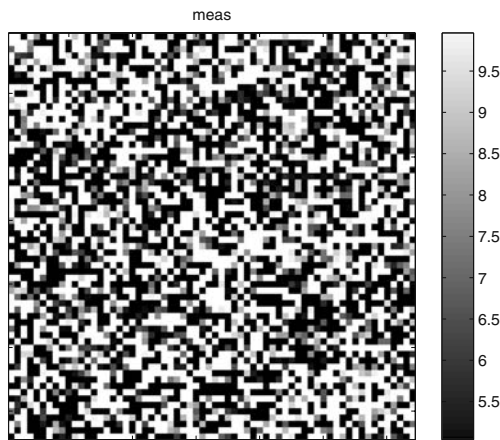
Fig. 12 The data

Fig. 14 A row-wise numbering of the pixels in the numerical grid

1	2	3	4	5	6	⋯
65	66	67	68	69	70	⋯
129	130					

⋮

quadtree (four children for every parent), there are 16 pixels assigned to each finest-scale node, and 16 states preserved at coarser scale nodes.

The task is to assimilate large-scale data. It might be possible to obtain the measurement in every pixel of the field. This kind of data is very noisy and, obviously, the number of data points is very large. It is known that EnKF is not an efficient tool to assimilate a very large amount of observations. The standard deviation of the measurement noise is, therefore, equal to a large value of 9. The data are shown in Fig. 12.

The EnMSF will be run twice, each time with a different grid numbering. The numbering schemes are shown in Figs. 13 and 14.

The numbering can express our belief in the dependencies in the actual field. The square-manner numbering (Fig. 13) keeps groups of pixels close in the grid close in the tree. It is not a perfect mapping, though. For example, pixels 6 and 17 are direct neighbors but they are placed at different nodes.

The other approach (Fig. 14) numbers the pixels row-wise as if one believes that the channels are horizontal. It can be improved if there is some prior knowledge available, for example, about the channel placement. It is visible in the results that interesting artifacts come from those two different approaches.

The plots of the prior, EnKF estimation and EnMSF with square and row-wise numbering estimations are

shown in Figs. 15, 16, 17, and 18. The prior is relatively smooth and it is the best estimate if no data are given (the mean of the ensemble). Any proper assimilation should give an improvement to the prior, which is the case in here.

The comparison of the performances is based on a root mean square error (RMSE) values and visual judgment. Table 1 contains RMSE between the truth and the prior, EnKF, EnMSF + square numbering, EnMSF + row-wise numbering.

The RMSE measures, roughly, the mean difference between respective pixels. It is a point not-global measure; it cannot give information on large-scale features. Additionally, one update step should not only rely on the RMSE. Hence, the visual comparison is also useful. It might suggest a need to search for a completely different measure of similarity.

The plot of EnKF in Fig. 16 is smooth, and it seems like it sharpens a contrast in the prior. Its RMSE is not satisfactory either. The two versions of EnMSF (Figs. 17 and 18) show artifact lines, which come from the numbering schemes used. Nevertheless, the plots extract the high-permeability channels quite well. The two approaches were going to show that EnMSF can be adjusted to a given problem, especially when some prior knowledge is available about the channel orientation or concentration.
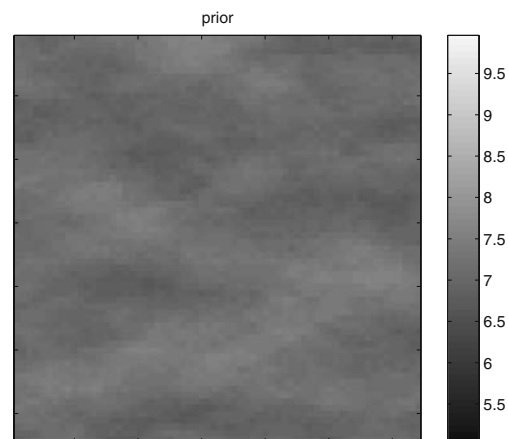
Fig. 13 A square-manner numbering of the pixels in the numerical grid

1	2	5	6	17	18	⋯
3	4	7	8	19	20	
9	10	13	14			
11	12	15	16			

⋮

Fig. 15 A mean of the ensemble members—the prior
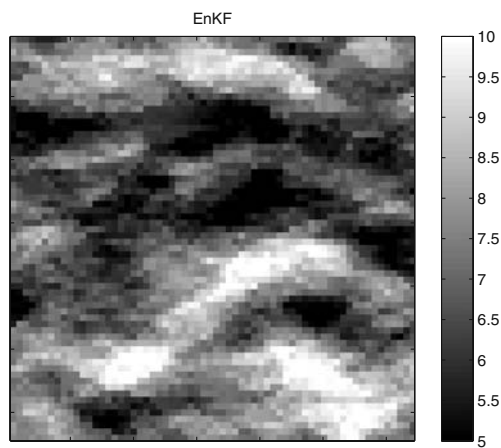
Fig. 16 Assimilation with EnKF
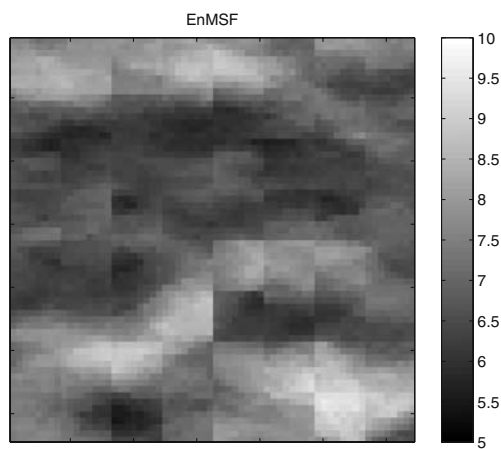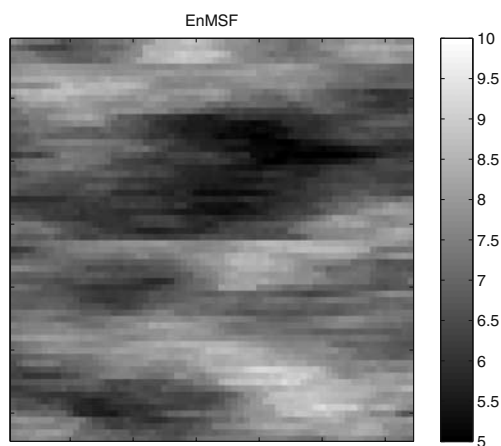
Table 1 RMSE between the truth and different results

Prior	EnKF	EnMSF+square	EnMSF+row wise
1.4002	1.3356	1.0795	1.0773

4 Conclusions

The EnMSF is a new technique for reservoir engineering. The method has been developed from image processing. The goal of this paper is to show an application of this filter to a simple reservoir engineering problem and to analyze its potential.

It is known that large data sets cause computational problems for Kalman filters. Therefore, there is a need for efficient tools to handle this kind of application.

Multiscale filtering is a way of representing the covariance matrix in the assimilation process by a tree structure. This simplification preserves the strongest correlations between the grid cells. The most complicated part of the method is the definition of the tree; it contains crucial assumptions and flexible parameters. There are features that influence the filter's performance that can be adjusted to solve particular problems. Here, we focused on the numbering schemes that can represent our belief in the field dependencies. Certainly, it is very efficient to manipulate when some prior knowledge about the field is available.

The two numbering schemes shown in this paper represent different ideas. The first one, square-like, might be universal to keep close pixels on the grid close in the tree. The second, row wise, can be suggested by horizontal flow information. Both schemes show good performance compared to EnKF in case of large data sets. The perfect mixture would be created when an approximate position of the channel was known. The shape or way of numbering could be adjusted to the feature.

Since the EnMSF is a complex and interesting algorithm, it needs further experiments and investigation. Full runs with a reservoir simulator and more tests are required.

Fig. 17 Assimilation with EnMSF and numbering scheme like in Fig. 13

Appendix A

The proper detailed description of the idea is presented in [29]. We search for a set of $V(s)$ values that provides *the scale-recursive Markov property* on the tree, i.e., decorrelates $q + 1$ following sets of one scale: first, q sets are all the children of the node s, and the set $q + 1$ contains all the other nodes in this scale that are not children of s. The decorrelation is a minimization of

Fig. 18 Assimilation with EnMSF and numbering scheme like in Fig. 14

conditional cross-covariances between the mentioned sets, given node s.

The tree that will approximate the forecast covariance matrix well should be based on *the scale-recursive Markov property*. The set of $V(s)$ values providing the scale-recursive Markov property perfectly would have a very high dimension since it would keep the total dependence between the finest states on the upper scale. Therefore, for practical purposes, the state dimensions in coarser scales will be constrained. This is easier if $V(s)$ values are *block diagonal*; each block corresponds to a different and only one child of s.

The way $V(s)$ values are built $V(s)$ has the form:

$$V(s) = diag[V_1(s), ..., V_q(s)],$$

where $V_i(s)$ is a matrix corresponding to the *i*th child of s, $s\alpha_i$, for $i = 1, ..., q$.

There are two constraints hidden here. The first one limits the number of rows in matrices $V_i(s)$ to $d_i(s)$. The second one, if necessary, coarsens the number of rows in $V(s)$.

Constructing matrices $V_i(s)$ To obtain $V_i(s)$ values, q conditional covariances would have to be minimized for each non-fine-scale node s. Those would be the conditional cross-covariances between child i ($i = 1, ..., q$) and the rest of the nodes in the same scale, given the

parent. Since direct minimization is inconvenient, the algorithm uses a *predictive efficiency method*.

Predictive efficiency method The method is more efficient to compute than all the conditional cross-covariances. It picks $V_i(s)$ values that minimize the departure from optimality of the estimate:

$$\hat{z}_{ic}(s) = E[z_{ic}(s)|V_i(s)z_i(s)],$$

where $z_i(s)$ is a vector of states at node $s\alpha_i (= \chi(s))$ and $z_{ic}(s)$ is a vector of states on all nodes at scale $m(s) + 1$ except node $s\alpha_i$. It was proven [7] that they are given by the first $d_i(s)$ rows of:

$$V_i'(s) = U_i^T(s)Cov[z_i(s)]^{-1/2},$$

where $U_i(s)$ contains the column eigenvectors of:

$$Cov^{-1/2}[z_i(s)]Cov[z_i(s), z_{ic}(s)]Cov^T$$
$$\times [z_i(s), z_{ic}(s)]Cov^{-T/2}[z_i(s)].$$

Here, it should be noted that $d_i(s)$ are chosen by the user. The picked rows have the highest corresponding eigenvalues. The reason is that we assume that the column eigenvectors of $U_i(s)$ are lined in a decreasing (corresponding eigenvalue) order.

Appendix B

$$K(s) = \widehat{Cov}[\chi(s), \hat{Y}(s)][\widehat{Cov}[\hat{Y}(s)] + R(s)]^{-1}$$

$$\begin{cases} R(s) = r(s), & m(s)=M; \\ R(s) = diag[K(s\alpha_1)R(s\alpha_1)K^T(s\alpha_1), ..., K(s\alpha_q)R(s\alpha_q)K^T(s\alpha_q)], & m(s)<M. \end{cases}$$

$$\begin{cases} Y^j(s) = y(s) + e^j(s), & m(s)=M; \\ Y(s) = \begin{bmatrix} K(s\alpha_1)Y^j(s\alpha_1) \\ \vdots \\ K(s\alpha_q)Y^j(s\alpha_q) \\ y(s) + e^j(s) \end{bmatrix}, & m(s)<M. \end{cases}$$

$$\begin{cases} \hat{Y}(s) = h(s)\chi_M^j(s), & m(s)=M; \\ \hat{Y}(s) = \begin{bmatrix} K(s\alpha_1)\hat{Y}^j(s\alpha_1) \\ \vdots \\ K(s\alpha_q)\hat{Y}^j(s\alpha_q) \\ h(s)\chi_M^j(s) \end{bmatrix}, & m(s)<M. \end{cases}$$

Appendix C

$$J(s) = \widehat{Cov}[\chi(s|s)]F^T(s)\widehat{Cov}^{-1}[\chi(s\gamma|s)]$$

$$F(s) = \widehat{Cov}[\chi(s\gamma)]A(s)^T\widehat{Cov}^{-1}[\chi(s)]$$

$$A(s) = \widehat{Cov}[\chi(s), \chi(s\gamma)]\widehat{Cov}^{-1}[\chi(s\gamma)]$$

Appendix D

$$\chi^j(s\gamma|s) = F(s)\chi^j(s|s) + w'^j(s).$$

Matrix $F(s)$ is like that in Appendix C and $w'^j(s)$ is a zero-mean random perturbation with covariance $Q'(s)$:

$$Q'(s) = \widehat{Cov}[\chi(s\gamma)] - F(s)A(s)\widehat{Cov}[\chi(s\gamma)],$$

where $F(s)$ and $A(s)$ are like those in Appendix C.

References

1. Caers, J.: Petroleum Geostatistics. Society of Petroleum Engineers, Richardson (2005)
2. Courtier, P., Derber, J., Errico, R., Louis, J.-F., Vukicevic, T.: Important literature on the use of adjoint, variational methods and the kalman filter in meteorology. Tellus **45A**, 342 (1993)
3. Dee, D.P.: Simplification of the kalman filter for meteorological data assimilation. Q. J. R. Meteorol. Soc. **117**, 365–384 (1991)
4. Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. J. Geophys. Res. **99**, 10143–10162 (1994)
5. Evensen, G.: Sampling strategies and square root analysis schemes for the enkf. Ocean Dyn. **54**, 539–560 (2004)
6. Evensen, G., van Leeuwen, P.: Assimilation of geosat altimeter data for the agulhas current using ensemble kalman filter with quasi-geostrophic model. Mon. Weather Rev. **124**, 85–96 (1996)
7. Frakt, A., Willsky, A.: Computationally efficient stochastic realization for internal multiscale autoregressive models. Multidimens. Syst. Signal Process. **12**, 109–142 (2001)
8. Gao, G., Reynolds, A.: Quantifying uncertainty for the punq-s3 problem in a bayesian settings with rml and enkf. In: SPE Reservoir simulation symposium, SPE 93324, SPE 93324, The Woodlands, 31 January–2 Feburary 2005
9. Ghil, M., Malanotte-Rizzoli, P.: Data assimilation in meteorology and oceanography. Adv. Geophys. **33**, 141–266 (1991)
10. Gu, Y., Oliver, D.: History matching of the punq-s3 reservoir model using te ensemble kalman filter. In: SPE Annual Technical Conference and Exhibition, SPE 89942, Houston, 26–29 September 2004
11. Hamill, T., Whitaker, J., Snyder, C.: Distance-dependent filtering of background error covariance estimates in an ensemble kalman filter. Mon. Weather Rev. **129**, 2884–2903 (2001)
12. Hanea, R., Velders, G., Heemink, A.: Data assimilation of ground level ozone in Europe with a kalman filter and chemistry transport model. J. Geophys. Res. **109**, 1–19 (2004)
13. Houtekamer, P., Mitchell, H.L.: Data assimilation using an ensemble kalman filter technique. Mon. Weather Rev. **126**, 796–811 (1998)
14. Houtekamer, P., Mitchell, H.L.: A sequential ensemble kalman filter for atmospheric data assimilation. Mon. Weather Rev. **129**, 123–137 (2001)
15. Liu, N., Oliver, D.: Critical evaluation of the ensemble kalman filter on history matching of geoogical facies. In: SPE Reservoir Simulation Symposium, SPE 92867, The Woodlands, 31 January–2 Feburary 2005
16. Lorenc, A.: The potential for the ensemble kalman filter for nwp: a comparison with 4dvar. Q. J. R. Meteorol. Soc. **129**, 3183–3203 (2003)
17. Margulis, S., McLaughlin, D., Entekhabi, D., Dune, S.: Land data assimilation and estimation of soil moisture using measurements from the southern great plains 1997 field experiment. Water Resour. Res. **38**, 1299 (2002)
18. Mitchell, H.L., Houtekamer, P., Pellerin, G.: Ensemble size, balance, and model-error representation in an ensemble kalman filter. Mon. Weather Rev. **130**, 2791–2808 (2002)
19. Nævdal, G., Johnson, L., Aanonsen, S., Vefring, E.: Reservoir monitoring and continuos model updating using ensemble kalam filter. In: SPE Annual Technical Conference and Exhibition, Houston, 26–29 September 2004
20. Nævdal, G., Mannset, T., Vefring, E.: Instrumented wells and near well reservoir monitoring through ensemble kalman filter. In: 8th European Conference on the Mathematics of Oil Recovery (2002a)
21. Nævdal, G., Mannset, T., Vefring, E.: Near well reservoir monitoring through ensemble kalman filter. In: SPE 75235 (2002b)
22. Ott, E., Hunt, B., Szunyogh, I., Zimin, A., Kostelich, E., Corazo, M., Kalnay, E.D.P., Yorke, J.: A local ensemble kalman filter for atmospheric data assimilation. Tellus A, **56**, 415–428 (2004)
23. Segers, A., Heemink, A., M.Verlaan, van Loon, M.: A modified rrsqrt-filter for assimilating data in atmospheric chemistry models. Environ. Model. Softw. **15**, 663–671 (2000)
24. Skjervheim, J.A., Evensen, G., Aanonsen, S.I., Ruud, B.O., Johansen, T.A.: Incorporating 4d seismic data in reservoir simulation models using ensemble kalman filter. In: SPE Annual Technical Conference and Exhibition, SPE 95789 (2005)
25. Verlaan, M., Heemink, A.: Tidal flow forecasting using reduced-rank square root filters. Stoch. Hydrol. Hydraul. **11**, 349–368 (1997)
26. Wang, K., Lary, D., Shallcross, D., Hall, S., Pyle, J.: A review on the use of the adjoint method in four-dimensional atmospheric-chemistry data assimilation. Q. J. R. Meteorol. Soc. **127**, 2181–2204 (2001)
27. Wen, X.H., Chen, W.H.: Real-time reservoir model updating using ensemble kalman filter. In: SPE Reservoir Simulation Symposium, SPE 92991 (2005)
28. Willsky, A.: Multiresolution markov models for signal and image processing. Proc. I. E. E. E. **90**, 1396–1458 (2002)
29. Zhou, Y., McLaughlin, D., Entekhabi, D., Ng, G.: An ensemble multiscale filter for large nonlinear data assimilation problems. Mon. Weather. Rev. **136**, 678–698 (2008)