Computers & Geosciences 55 (2013) 54–69

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Reduced-order flow modeling and geological parameterization for ensemble-based data assimilation

Jincong He [a,*], Pallav Sarma [b], Louis J. Durlofsky [a]

[a] Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305, USA
[b] Chevron Energy Technology Company, 6001 Bollinger Canyon, San Ramon, CA 94583, USA

ARTICLE INFO

Article history:
Received 13 January 2012
Received in revised form
8 March 2012
Accepted 30 March 2012
Available online 7 April 2012

Keywords:
Trajectory piecewise linearization
Reservoir simulation
History matching
Reduced-order model
Karhunen–Loève expansion
Ensemble Kalman filter

ABSTRACT

Reduced-order modeling represents an attractive approach for accelerating computationally expensive reservoir simulation applications. In this paper, we introduce and apply such a methodology for data assimilation problems. The technique applied to provide flow simulation results, trajectory piecewise linearization (TPWL), has been used previously for production optimization problems, where it has provided large computational speedups. The TPWL model developed here represents simulation results for new geological realizations in terms of a linearization around previously simulated (training) cases. The high-dimensional representation of the states is projected into a low-dimensional subspace using proper orthogonal decomposition. The geological models are also represented in reduced terms using a Karhunen–Loève expansion of the log-transmissibility field. Thus, both the reservoir states and geological parameters are described very concisely. The reduced-order representation of flow and geology is appropriate for use with ensemble-based data assimilation procedures, and here it is incorporated into an ensemble Kalman filter (EnKF) framework to enrich the ensemble at a low cost. The method is able to reconstruct full-order states, which are required by EnKF, whenever necessary. The combined technique enables EnKF to be applied using many fewer high-fidelity reservoir simulations than would otherwise be required to avoid ensemble collapse. For two- and three-dimensional example cases, it is demonstrated that EnKF results using 50 high-fidelity simulations along with 150 TPWL simulations are much better than those using only 50 high-fidelity simulations (for which ensemble collapse is observed) and are, in fact, comparable to the results achieved using 200 high-fidelity simulations.

1. Introduction

History matching is an essential component of reservoir modeling and management. It entails the updating of reservoir models using dynamic data, e.g., production data or time-lapse seismic data. History matching typically requires large numbers of simulations, so it can be extremely time consuming if high-resolution models are used. Thus, there is a significant need for efficient (proxy or surrogate) models that can predict simulation results with reasonable accuracy.

Reduced-order modeling procedures, which have been applied in many application areas including reservoir simulation, represent a promising means for constructing efficient surrogate models. Many of these techniques entail the projection of the full-order (high-fidelity) numerical description into a low-dimensional subspace, which reduces the number of unknowns that must be computed at each time step. Existing approaches, applied within the context of reservoir simulation, include procedures based on proper orthogonal decomposition or POD (van Doren et al., 2006; Cardoso et al., 2009) and techniques based on the combination of POD and trajectory piecewise linearization, TPWL (Cardoso and Durlofsky, 2010; He et al., 2011a). The target application in these studies was mainly production optimization, and the reduced-order model was used to provide results for varying well control parameters (e.g., bottomhole pressure).

In this work, we explore the use of the TPWL reduced-order modeling approach for data assimilation. In TPWL, new solutions are represented as linearizations around saved states from previously simulated (training) solutions. The linearized representation is projected into a low-dimensional space using POD. TPWL was first introduced by Rewienski and White (2003) and has since been applied in a number of areas (Gratton and Willcox, 2004; Yang and Shen, 2005; Vasilyev et al., 2006) including reservoir simulation (as noted above). As far as we are aware, the use of TPWL (or a closely related procedure) for data assimilation has not been previously explored.

* Corresponding author.
E-mail address: jincong@stanford.edu (J. He).

0098-3004/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cageo.2012.03.027

The approach applied here entails the use of geological variables as control parameters within the TPWL representation. This enables us to perform training runs for particular geological models and to then represent solutions for new geological models as linearized expansions around the training runs. The geological model itself can also be represented in a reduced space, again through the use of POD. When applied within the context of geological modeling, this reduction is typically referred to as principal component analysis (PCA) or Karhunen–Loève (K–L) expansion; see Oliver (1996) and Sarma et al. (2006, 2008) for further discussion. Thus, our overall framework entails low-order representations of both the reservoir states (pressure and saturation) and the geological parameters (transmissibility in our case).

In this study, the TPWL representation is incorporated into an ensemble Kalman filter (EnKF) framework. EnKF was introduced by Evensen (1994) and has since become very popular in the petroleum industry. Nævdal et al. (2002) first used EnKF to history match near-well static parameters. In subsequent work, EnKF was applied to update the entire reservoir model by continuously adjusting the permeability, pressure and saturation fields (Nævdal et al., 2005). The procedure has been successfully tested on several field cases including the PUNQ-S3 reservoir model (Gu and Oliver, 2005). EnKF has also been integrated into a closed-loop reservoir modeling framework (Nævdal et al., 2006). See Aanonsen et al. (2009) for a detailed review of recent developments in EnKF. Although EnKF provides multiple realizations that approximately match historical data, some concerns regarding its ability to properly quantify uncertainty are expressed by Park (2011).

Despite its success, a well-known limitation of EnKF is that, when the ensemble size is small, the ensemble approximation to the cross-covariance matrix can be spurious, resulting in unphysical updates of reservoir properties and loss of ensemble variability. This problem is referred to as 'ensemble collapse' (Lorenc, 2009). In addition, the number of degrees of freedom in the ensemble is limited by the ensemble size. The ensemble collapse problem is typically addressed through use of localization, where the estimated covariance is modified to avoid spurious correlations and reduce collinearity (Zhang and Oliver, 2010; Chen and Oliver, 2010). Localization is, however, essentially a heuristic fix and is not always reliable. In addition, even when it performs adequately, localization only addresses ensemble collapse, not the issue of limited degrees of freedom.

The problems noted above, which result from the use of a small ensemble, do not exist if we can afford to use a sufficiently large ensemble. In a typical EnKF implementation, however, the total computational time scales directly (linearly) with the number of realizations in the ensemble, so the use of large ensembles can be expensive. In this work, we use the TPWL representation for the majority of realizations. This provides us with much of the benefit of a large ensemble (e.g., avoidance of ensemble collapse), but at reduced cost.

This paper proceeds as follows. We first describe the TPWL representation of flow solutions for varying geological parameters. We also discuss the representation of the geological model using the K–L expansion. Results demonstrating the use of the overall reduced-order modeling procedure for computing the production response of new geological realizations are then presented. Next, we give a brief overview of EnKF and describe the combined EnKF-TPWL workflow. The use of the EnKF-TPWL methodology for two- and three-dimensional systems is then demonstrated. We conclude with a summary and a discussion of outstanding issues and future research directions.

An earlier version of this work, including some of the discussion contained here, was presented in an SPE conference proceedings paper (He et al., 2011b). The descriptions and results in this paper are, however, further developed and much more comprehensive than those in He et al. (2011b).

2. TPWL formulation for variable geological parameters

In this section we describe the TPWL formulation with variable geological parameters in detail. The representation of geological parameters using the K–L expansion is also discussed. We limit our descriptions to oil–water systems, as the TPWL method has not yet been implemented for general three-phase black-oil models. Extensions along these lines will be considered in future work.

2.1. Discretized flow equations

Detailed derivations of TPWL models for oil–water systems, with fixed geological parameters but varying well controls (bottomhole pressure), have been presented previously by Cardoso and Durlofsky (2010) and He et al. (2011a). Our discussion here follows the descriptions in these references.

The partial differential equations governing subsurface flow are derived by combining statements of mass conservation for oil and water with Darcy's law for each phase. Discrete representations, based on standard finite-volume treatments and fully implicit schemes, are then introduced (see Cardoso and Durlofsky, 2010 for details). The discretized system, which is a set of nonlinear algebraic equations, can be written as

$$\mathbf{g}(\mathbf{x}^{n+1},\mathbf{x}^{n},\mathbf{m})=0, \tag{1}$$

where $\mathbf{g}$ is the residual we seek to drive to zero, n and $n+1$ designate time levels, $\mathbf{x}$ is the state vector (the states are pressure p and water saturation S_w in each grid block), and $\mathbf{m}$ represents the geological parameters to be determined by history matching. For a model containing N_c grid blocks, $\mathbf{g}$ and $\mathbf{x}$ are vectors of dimension $2N_c$. If $\mathbf{m}$ is taken to represent directional permeability (k_x, k_y, k_z) in each grid block in a three-dimensional problem, it is of dimension $3N_c$.

In reservoir simulation, Eq. (1) is nonlinear and is typically solved using Newton's method. Practical problems may have on the order of 10^5 or 10^6 unknowns, which makes the solution expensive computationally (though much smaller models will be considered in this work). We now describe the application of the TPWL approach for the efficient solution of Eq. (1).

2.2. Piecewise linearization of the discretized equations

In trajectory piecewise linearization (TPWL), we first simulate some number of full-order 'training' runs using particular geological models. We designate the geological model used for the training run ω as $\mathbf{m}_\omega$. This training information is then applied to build a TPWL surrogate. The surrogate can then be used for a simulation with a different geological description, which we designate as $\mathbf{m}$.

We proceed by linearizing Eq. (1) around states saved during training simulations. Here, at any given time, we linearize around a single point (rather than using weighted linearizations around multiple points) on the training trajectory. Given the geological model $\mathbf{m}$ and the (known) state at the current time $\mathbf{x}^n$, we designate the 'closest' saved state, which is the state at time step i in the training run for geological model $\mathbf{m}_\omega$, as $\mathbf{x}^i_\omega$. We will explain later (in Section 2.6) how $\mathbf{m}_\omega$ and i are found. To determine the new state $\mathbf{x}^{n+1}$, we linearize $\mathbf{g}$ around

$(\mathbf{x}_\omega^{i+1},\mathbf{x}_\omega^i,\mathbf{m}_\omega)$. This gives

$$\mathbf{g}^{n+1} \approx \mathbf{g}_\omega^{i+1} + \frac{\partial \mathbf{g}_\omega^{i+1}}{\partial \mathbf{x}_\omega^{i+1}}(\mathbf{x}^{n+1}-\mathbf{x}_\omega^{i+1}) + \frac{\partial \mathbf{g}_\omega^{i+1}}{\partial \mathbf{x}_\omega^{i}}(\mathbf{x}^{n}-\mathbf{x}_\omega^{i}) + \frac{\partial \mathbf{g}_\omega^{i+1}}{\partial \mathbf{m}_\omega}(\mathbf{m}-\mathbf{m}_\omega), \tag{2}$$

where $\mathbf{g}^{n+1}=\mathbf{g}(\mathbf{x}^{n+1},\mathbf{x}^n,\mathbf{m})=0$ and $\mathbf{g}_\omega^{i+1}=\mathbf{g}(\mathbf{x}_\omega^{i+1},\mathbf{x}_\omega^i,\mathbf{m}_\omega)=0$ from Eq. (1). Note that the $\mathbf{m}$ vector appearing here will be specified by the history matching algorithm, so it is a known quantity. The derivatives in Eq. (2) will be denoted as

$$\mathbf{J}_\omega^{i+1} = \frac{\partial \mathbf{g}_\omega^{i+1}}{\partial \mathbf{x}_\omega^{i+1}}, \quad \mathbf{A}_\omega^{i+1} = \frac{\partial \mathbf{g}_\omega^{i+1}}{\partial \mathbf{x}_\omega^{i}}, \quad \mathbf{B}_\omega^{i+1} = \frac{\partial \mathbf{g}_\omega^{i+1}}{\partial \mathbf{m}_\omega}. \tag{3}$$

Note that $\mathbf{J}_\omega^{i+1}$ is the Jacobian matrix at time step $i+1$ of training simulation ω (i.e., the training run for geological model $\mathbf{m}_\omega$). Eq. (2) can now be expressed as

$$\mathbf{J}_\omega^{i+1}(\mathbf{x}^{n+1}-\mathbf{x}_\omega^{i+1}) = -[\mathbf{A}_\omega^{i+1}(\mathbf{x}^{n}-\mathbf{x}_\omega^{i}) + \mathbf{B}_\omega^{i+1}(\mathbf{m}-\mathbf{m}_\omega)]. \tag{4}$$

For conciseness, from here on we will drop the ω subscript in $\mathbf{x}_\omega^i$, $\mathbf{x}_\omega^{i+1}$, $\mathbf{J}_\omega^{i+1}$, $\mathbf{A}_\omega^{i+1}$, $\mathbf{B}_\omega^{i+1}$ and $\mathbf{g}_\omega^{i+1}$, although it will be retained in $\mathbf{m}_\omega$. However, it should be understood that terms with superscripts i and $i+1$ correspond to training simulation ω.

During the training simulation, we save $\mathbf{x}^i$, $\mathbf{J}^{i+1}$, $\mathbf{A}^{i+1}$ and $\mathbf{B}^{i+1}$ at each time step. Given $\mathbf{x}^n$ and this saved information, Eq. (4) provides a linearized representation for $\mathbf{x}^{n+1}$, which is the only unknown variable. This equation is, however, still in a high-dimensional space; i.e., the dimension of $\mathbf{J}^{i+1}$ is $2N_c \times 2N_c$. We now describe the POD procedure, which is used to project Eq. (4) to a reduced space.

2.3. Model order reduction by proper orthogonal decomposition

Through application of POD, we can represent the state vector $\mathbf{x}$ in terms of a reduced state vector $\mathbf{z}$ and a basis matrix $\mathbf{\Phi}$ using

$$\mathbf{x} \approx \mathbf{\Phi}\mathbf{z}. \tag{5}$$

POD, which is also referred to as principal component analysis or PCA, is a general procedure that uses an orthogonal transformation to represent a set of realizations of (possibly correlated) variables $\mathbf{x}$ in terms of uncorrelated variables $\mathbf{z}$ (Jolliffe, 2002). In practice, POD can be accomplished by performing singular value decomposition (SVD) of the so-called snapshot matrix of $\mathbf{x}$. The resulting singular vectors provide the orthonormal basis matrix $\mathbf{\Phi}$. POD is optimal in the sense that it minimizes the mean-square reconstruction error for the snapshots. Therefore, it is reasonable to assume that POD can represent the states of other (test) simulations if these states are somewhat similar to those encountered during training runs.

Our approach is as follows. We denote the total number of training simulations as N_ω and the total number of time steps from all of these N_ω training simulations as L. We save the full pressure and saturation states (snapshots) at each time step of each simulation. These snapshots are then assembled into pressure and saturation snapshot matrices. Designating the pressure snapshots as $\mathbf{x}_p^1$, $\mathbf{x}_p^2$, etc., and the saturation snapshots as $\mathbf{x}_s^1$, $\mathbf{x}_s^2$, etc., the snapshot matrices are constructed as

$$\mathbf{X}_p = [\mathbf{x}_p^1 \ \mathbf{x}_p^2 \ \ldots \ \mathbf{x}_p^L], \quad \mathbf{X}_s = [\mathbf{x}_s^1 \ \mathbf{x}_s^2 \ \ldots \ \mathbf{x}_s^L]. \tag{6}$$

The columns of the orthonormal basis matrices $\mathbf{\Phi}_p$ and $\mathbf{\Phi}_s$ are given by the left singular vectors of the SVDs of $\mathbf{X}_p$ and $\mathbf{X}_s$, scaled by the corresponding principal values. There are a maximum of L nonzero singular values, though in general it is not necessary to retain all L columns in $\mathbf{\Phi}_p$ and $\mathbf{\Phi}_s$. These two matrices are then assembled into a single basis matrix $\mathbf{\Phi}$ as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{S}_w \end{bmatrix} \approx \mathbf{\Phi}\mathbf{z} = \begin{bmatrix} \mathbf{\Phi}_p & 0 \\ 0 & \mathbf{\Phi}_s \end{bmatrix} \begin{bmatrix} \mathbf{z}_p \\ \mathbf{z}_s \end{bmatrix}, \tag{7}$$

where $\mathbf{p}$ and $\mathbf{S}_w$ designate the vectors of pressure and saturation unknowns. The number of columns to be retained in $\mathbf{\Phi}_p$ and $\mathbf{\Phi}_s$ can be determined using either an energy criterion (Cardoso and Durlofsky, 2010) or through a basis optimization procedure (He et al., 2011a). Designating the number of columns in these matrices as l_p and l_s, the dimension of $\mathbf{\Phi}$ is $2N_c \times l$, where $l = l_p + l_s$. In general, $l \ll 2N_c$, so significant reduction is achieved.

2.4. Linearized reduced-order representation

We now combine the linearized representation in Eq. (4) with the POD reduction described above. Substituting Eq. (5) into Eq. (4), we obtain the following over-determined system ($2N_c$ equations, l unknowns):

$$\mathbf{J}^{i+1}\mathbf{\Phi}(\mathbf{z}^{n+1}-\mathbf{z}^{i+1}) = -[\mathbf{A}^{i+1}\mathbf{\Phi}(\mathbf{z}^{n}-\mathbf{z}^{i}) + \mathbf{B}^{i+1}(\mathbf{m}-\mathbf{m}_\omega)]. \tag{8}$$

To reduce the number of equations to equal the number of unknowns, we pre-multiply both sides of Eq. (8) by $\mathbf{\Phi}^T$, which is referred to as Galerkin projection (Berkooz and Titi, 1993). This gives, after further manipulation

$$\mathbf{z}^{n+1} = \mathbf{z}^{i+1} - (\mathbf{J}_r^{i+1})^{-1}[\mathbf{A}_r^{i+1}(\mathbf{z}^{n}-\mathbf{z}^{i}) + \mathbf{B}_r^{i+1}(\mathbf{m}-\mathbf{m}_\omega)], \tag{9}$$

where

$$\mathbf{J}_r^{i+1} = \mathbf{\Phi}^T\mathbf{J}^{i+1}\mathbf{\Phi}, \quad \mathbf{A}_r^{i+1} = \mathbf{\Phi}^T\mathbf{A}^{i+1}\mathbf{\Phi}, \quad \mathbf{B}_r^{i+1} = \mathbf{\Phi}^T\mathbf{B}^{i+1}. \tag{10}$$

Note that reduction methods other than the Galerkin projection exist; e.g., the least-square projection (Carlberg et al., 2011). It will be of interest to evaluate the performance of this and other approaches in future work.

In Eq. (9), all vector and matrix dimensions are reduced (to size l) except for $\mathbf{m}$ and $\mathbf{m}_\omega$ and the number of columns in $\mathbf{B}_r^{i+1}$, which are still of dimension $O(N_c)$. In order to render all terms in Eq. (9) in low-dimensional space, we need to introduce the reduced-order representation for the geological model. We now describe our K–L treatment.

2.5. Reduced-order representation of geological parameters

In this work, for convenience, we consider systems in which the permeability field varies from model to model but porosity is constant. There are several ways to represent the geological models $\mathbf{m}$ and $\mathbf{m}_\omega$ in Eq. (9) for such cases. Obvious choices are the directional permeabilities in each grid block (designated $\mathbf{k}$, of dimension $3N_c$) or transmissibility in each grid block ($\mathbf{T}$). Numerical experiments have demonstrated that, within the context of TPWL, the use of $\mathbf{T}$ is preferable to the use of $\mathbf{k}$, presumably because $\mathbf{g}$ is linear in $\mathbf{T}$ (which means one of the neglected higher-order terms in Eq. (2) vanishes), while it is nonlinear in $\mathbf{k}$. Because transmissibility is defined at interior block interfaces, the dimension of $\mathbf{T}$ (designated N_τ) is $3N_xN_yN_z - N_xN_y - N_yN_z - N_xN_z$ for Cartesian grids, where N_x, N_y and N_z are the number of grid blocks in each direction.

We will apply a K–L representation (which is, to reiterate, completely analogous to the POD representation described earlier) to express $\mathbf{T}$ in a reduced form. When using the standard two-point flux approximation for flow terms, as is applied here, all transmissibilities must be positive to ensure flow is 'down' the pressure gradient. However, direct use of the K–L representation to provide new $\mathbf{T}$ fields does not guarantee the positivity of all components of $\mathbf{T}$. For this reason, instead of working directly with transmissibility, we use log-transmissibility, designated $\boldsymbol{\eta}$. This guarantees that the K–L derived transmissibility values are always positive.

We follow the basic procedure described in Sarma et al. (2006, 2008) to construct the K–L representation of $\boldsymbol{\eta}$. Given a particular variogram model and associated parameters (which define the

spatial correlation structure of permeability), along with any conditioning data, we apply SGEMS (Remy et al., 2008) to generate a set of N_r realizations of $\mathbf{k}$. From $\mathbf{k}$, $\mathbf{T}$ is constructed using standard procedures (e.g., in the x-direction we have $T_{ij}=2k_ik_jA/[(k_i+k_j)\Delta x]$, where k_i and k_j are the x-direction permeabilities of blocks i and j, respectively, A is the area of the common interface, and Δx is the block size), after which $\boldsymbol{\eta}$ is formed by taking the logarithm of each component of $\mathbf{T}$. Here, N_r is typically $O(10^2-10^3)$.

Each of the N_r realizations of $\boldsymbol{\eta}$ represents a snapshot, so a snapshot matrix $\boldsymbol{\Gamma}=[\boldsymbol{\eta}_1\ \ldots\ \boldsymbol{\eta}_{N_r}]$ can be constructed. Following Sarma et al. (2006, 2008), prior to constructing $\boldsymbol{\Gamma}$, the realizations are centered; i.e., $\overline{\boldsymbol{\eta}}=(1/N_r)\sum_{i=1}^{N_r}\boldsymbol{\eta}_i$ is subtracted from each $\boldsymbol{\eta}_i$. Next, following the procedure described in Section 2.3, we perform SVD on $\boldsymbol{\Gamma}$. Assembling the scaled left singular vectors in a basis matrix (designated $\mathbf{R}$), we can now express realizations of $\boldsymbol{\eta}$ as

$$\boldsymbol{\eta}=\mathbf{R}\boldsymbol{\xi}+\overline{\boldsymbol{\eta}}, \tag{11}$$

where $\mathbf{R}$ is of dimensions $N_\tau\times l_m$ ($l_m\le N_r$ is determined using an energy criterion) and $\boldsymbol{\xi}$ is the reduced variable.

We note that this K–L representation is most appropriate for use with Gaussian fields. It will entail some error for non-Gaussian (e.g., channelized) models, which are characterized by multipoint spatial statistics. A more general representation, referred to as kernel principal component analysis or KPCA, was developed in Sarma et al. (2008) to enable reduced-order representation of models characterized by multipoint statistics. Extension of the current framework to handle KPCA permeability representations may be considered in future work.

Given that each component of $\boldsymbol{\eta}$ is the logarithm of the corresponding component of $\mathbf{T}$, which we express as $\boldsymbol{\eta}=\log\mathbf{T}$, we have

$$\frac{\partial\mathbf{g}^{i+1}}{\partial\boldsymbol{\eta}_\omega}(\boldsymbol{\eta}-\boldsymbol{\eta}_\omega)=\frac{\partial\mathbf{g}^{i+1}}{\partial\mathbf{T}_\omega}\mathbf{D}_{T_\omega}(\boldsymbol{\eta}-\boldsymbol{\eta}_\omega), \tag{12}$$

where $\mathbf{T}_\omega$ is the vector of transmissibilities for training simulation ω (analogous to $\mathbf{m}_\omega$), $\boldsymbol{\eta}_\omega=\log\mathbf{T}_\omega$ and $\mathbf{D}_{T_\omega}$ is a diagonal matrix whose diagonal elements are the elements of $\mathbf{T}_\omega$. Using Eqs. (11) and (12), (9) can now be written as

$$\mathbf{z}^{n+1}=\mathbf{z}^{i+1}-(\mathbf{J}_r^{i+1})^{-1}[\mathbf{A}_r^{i+1}(\mathbf{z}^n-\mathbf{z}^i)+\tilde{\mathbf{B}}_r^{i+1}(\boldsymbol{\xi}-\boldsymbol{\xi}_\omega)], \tag{13}$$

with $\tilde{\mathbf{B}}_r^{i+1}$ given by

$$\tilde{\mathbf{B}}_r^{i+1}=\boldsymbol{\Phi}^T\frac{\partial\mathbf{g}^{i+1}}{\partial\mathbf{T}_\omega}\mathbf{D}_{T_\omega}\mathbf{R}. \tag{14}$$

This matrix is of dimension $l\times l_m$, where (to reiterate) l is the number of columns in $\boldsymbol{\Phi}$ and l_m is the number of columns in $\mathbf{R}$. Note that, although both l and l_m are small compared to N_c and N_τ, in general $l\ne l_m$.

Eq. (13) defines our TPWL model. This equation is linear and all the vectors and matrices appearing in it are in low-dimensional space, so the new reduced state $\mathbf{z}^{n+1}$ can be computed very efficiently. The full-order state $\mathbf{x}^{n+1}$ can be reconstructed from $\mathbf{z}^{n+1}$, at some or all grid blocks, using Eq. (5). The well production and injection rates are calculated from the well-block states using the standard well model, as described by Cardoso and Durlofsky (2010).

2.6. Point selection strategy

In order to use Eq. (13), we must first determine the geological model ($\mathbf{T}_\omega$) and associated saved states ($\mathbf{z}^i$, $\mathbf{z}^{i+1}$) to expand around. The determination of the geological model would not be required if we used only one training simulation, but the predictive ability of the resulting TPWL model in this case is highly limited. Through use of multiple training runs with different geological models, the TPWL basis incorporates information from a variety of geological scenarios. This greatly improves TPWL model accuracy and robustness for ensemble-based data assimilation problems.

When the TPWL model is constructed using multiple training simulations, at each time step of a new (test) simulation we need to choose, from all the saved training points (time steps) of all N_ω training simulations, a particular training point to linearize around. This point should be 'close' to the current point to minimize linearization error, although the precise definition of 'close' is problem dependent. Our procedure, established based on numerical experimentation, is as follows. We define a 'distance' between the current solution (at time step n in the test simulation) and the saved solution at time step j in training simulation v. This distance incorporates the difference between well-block pressures, well-block relative permeabilities, and near-well transmissibility, as these have a major impact on flow rate calculations.

The distance between the current solution and saved point j in training simulation v is defined as

$$d^{n,v,j}=\alpha d_p^{n,v,j}+\beta(d_{kro}^{n,v,j}+d_{krw}^{n,v,j})+\gamma d_T^{v}, \tag{15}$$

where α, β and γ are weights and $d_p^{n,v,j}$, $d_{kro}^{n,v,j}$ and d_T^v are the relative differences in well-block pressure, relative permeability and transmissibility between the current state (at time step n) and saved point j in training simulation v. These quantities are defined as

$$d_p^{n,v,j}=\frac{1}{N_s}\sum_{s=1}^{N_s}\left|\frac{p_{b,s}^n-p_{b,s}^{v,j}}{p_{b,s}^{v,j}-p_{w,s}^{v,j}+\epsilon}\right|,\quad d_{kro}^{n,v,j}=\frac{1}{N_s}\sum_{s=1}^{N_s}\left|\frac{k_{ro,s}^n-k_{ro,s}^{v,j}}{k_{ro,s}^{v,j}+\epsilon}\right|,$$
$$d_T^v=\frac{1}{N_\lambda}\sum_{\lambda=1}^{N_\lambda}\left|\frac{T_\lambda-T_\lambda^v}{T_\lambda^v+\epsilon}\right|, \tag{16}$$

where $p_{b,s}$ and $p_{w,s}$ are the well-block pressure and wellbore pressure in well-block s, $k_{ro,s}$ is the oil-relative permeability at well-block s, N_s is the total number of well blocks, and ϵ is a constant set to 0.01 to avoid dividing by zero. The quantities T_λ and T_λ^v are the transmissibility of the connection λ between well blocks and neighboring blocks for the test case and training simulation v, and N_λ is the total number of such connections. The expression for $d_{krw}^{n,v,j}$ is analogous to that for $d_{kro}^{n,v,j}$.

At each time step, Eq. (15) is evaluated for all training points (looping over v and j). The v and j pair that minimizes $d^{n,v,j}$ defines ω and i, which in turn provide $\mathbf{z}^i$ and the other vectors and matrices in Eq. (13). The reduced-order state for the subsequent time step (for the same ω) provides $\mathbf{z}^{i+1}$.

In our study, we use $\alpha=1$, $\beta=1$ and $\gamma=3$. The use of a relatively large value for γ avoids frequent shifts in ω; i.e., we tend to linearize around the same geological model over multiple time steps. Algorithm performance does not appear to be very sensitive to the choices for α, β and γ over a reasonable range.

2.7. Summary of the TPWL procedure

The overall methodology first requires pre-processing (offline) computations in which the TPWL model is constructed. The inline process provides TPWL results for each new (test) simulation model. These two TPWL components are summarized as follows:

Offline processing:

1. Generate N_r permeability realizations (using, e.g., SGEMS, described in Remy et al., 2008), the associated log-transmissibility vectors $\boldsymbol{\eta}$, and the 'snapshot' matrix $\boldsymbol{\Gamma}$. The SVD of $\boldsymbol{\Gamma}$ provides the K–L basis matrix $\mathbf{R}$.

2. Perform high-fidelity training simulations on N_ω realizations from the distribution considered in step 1. Save the states $\mathbf{x}^i$ and derivative terms $\mathbf{J}^i$, $\mathbf{A}^i$ and $\mathbf{B}^i$ at each time step in each run.
3. Construct the basis matrix $\mathbf{\Phi}$ by performing SVDs on the snapshot matrices $\mathbf{X}_p$ and $\mathbf{X}_s$.
4. Compute the reduced states $\mathbf{z}^i$ and reduced derivative matrices $\mathbf{J}_r^i$, $\mathbf{A}_r^i$ and $\tilde{\mathbf{B}}_r^i$.

Inline processing:

1. Given $\mathbf{z}^n$, find the training point (v,j) that minimizes $d^{n,v,j}$ in Eq. (15). This provides $\mathbf{z}^i$, $\mathbf{z}^{i+1}$ and other vectors and matrices to be used in the linearization.
2. Evaluate Eq. (13) to provide $\mathbf{z}^{n+1}$.
3. Reconstruct $\mathbf{x}^{n+1}$ from $\mathbf{z}^{n+1}$ at required locations using Eq. (5). Calculate well flow rates (using well-block states and standard well model) and any other required information.
4. Repeat steps 1–3 until the final simulation time is reached.

2.8. Implementation details

In many reservoir simulation applications such as history matching and production optimization, we are particularly interested in the pressure and saturation in well blocks because these values directly affect injection and production rates. Therefore, in our TPWL simulations here, we use the local-resolution scheme introduced in He et al. (2011a) to maintain high resolution in well blocks (other blocks may also be included at high resolution using a missing point estimation approach introduced by Astrid and Verhoeven, 2006). The local-resolution blocks are represented at full order; i.e., they are not projected into the low-order space using POD. We let $\mathbf{x}_{LR}$ designate the full-order states for the n_{LR} locally-resolved grid blocks (which we wish to represent at high resolution) and $\mathbf{x}_G$ are the states for the remaining (global) grid blocks. Then, instead of Eq. (5), we write

$$\mathbf{x} = \mathbf{P}\begin{bmatrix}\mathbf{x}_{LR}\\ \mathbf{x}_G\end{bmatrix} \approx \mathbf{P}\begin{bmatrix}\mathbf{\Phi}_{LR} & 0\\ 0 & \mathbf{\Phi}_G\end{bmatrix}\begin{bmatrix}\mathbf{z}_{LR}\\ \mathbf{z}_G\end{bmatrix} = \tilde{\mathbf{\Phi}}\begin{bmatrix}\mathbf{z}_{LR}\\ \mathbf{z}_G\end{bmatrix}, \tag{17}$$

where $\mathbf{P}$ is a permutation matrix that reorders $\mathbf{x}$ into $\mathbf{x}_{LR}$ and $\mathbf{x}_G$. The basis matrix $\mathbf{\Phi}_G$ is generated by application of POD to snapshots of $\mathbf{x}_G$, and $\mathbf{\Phi}_{LR}$ is simply the identity matrix. This gives $\mathbf{z}_{LR} = \mathbf{x}_{LR}$, meaning that full resolution is maintained for grid blocks associated with $\mathbf{x}_{LR}$. The modified basis matrix $\tilde{\mathbf{\Phi}}$ is then used in place of $\mathbf{\Phi}$ in Eq. (10). See He et al. (2011a) for further discussion.

The computational requirements for the TPWL model do not depend directly on the dimension of the original problem but rather on the dimension of the reduced representation. For a typical dimension of the reduced space ($l \sim 100-1000$) and a typical number of time steps ($\sim 50-300$), the runtime is on the order of one, or a few, seconds. However, the TPWL formulation does require performing training simulations and generating the low-dimensional states and matrices, which constitute most of the cost of the method (this is all done in the offline processing). The generation of the required matrices entails about as much computation time as the training runs.

In terms of storage, the full-order derivative terms $\mathbf{J}^i$, $\mathbf{A}^i$ and $\mathbf{B}^i$ (stored in step 2 of the offline processing) are sparse and are only intermediate products of the method. Their storage requirements can be reduced by more efficiently integrating the procedure into the simulator. The space requirements (step 4) for the reduced derivatives $\mathbf{J}_r^i$, $\mathbf{A}_r^i$ and $\tilde{\mathbf{B}}_r^i$, which are not sparse but are in low-dimensional space, are $O(L \times l^2)$ for $\mathbf{J}_r^i$ and $\mathbf{A}_r^i$ and $O(L \times l \times l_m)$ for $\tilde{\mathbf{B}}_r^i$, where L is the total number of time steps from all N_ω training runs. The storage requirement is proportional to l^2 or $l \times l_m$, although it does not depend directly on N_c, the dimension of the full-order model. For the examples considered in this work, about 10 Gb of storage are required. For significantly larger models, more efficient storage procedures may be needed.

The full-order simulations needed to generate the saved states and the derivative matrices are performed using Stanford's General Purpose Research Simulator, GPRS (Cao, 2002). GPRS has been modified to output the quantities required by TPWL at each time step of the training runs, which include $\mathbf{x}^i$, $\mathbf{J}^i$, $\mathbf{A}^i$ and $\partial \mathbf{g}^i / \partial \mathbf{T}_\omega$.

3. Evaluation of TPWL model performance

We now assess the performance of the TPWL model described in the preceding sections for an ensemble of realizations. These results motivate the use of this approach within an EnKF framework.

3.1. Geological realizations (Model 1)

This system (referred to as Model 1) is composed of two-dimensional geostatistical realizations defined on a 45×45 grid with cells of size 10 m $\times$ 10 m $\times$ 10 m. The realizations are generated using sequential Gaussian simulation within SGEMS (Remy et al., 2008). The $\log k$ fields are characterized by a Gaussian distribution with a mean of 5.0. Different values of the standard deviation of $\log k$ will be considered. A spherical variogram with ranges of 200 m and 100 m, oriented at 45° and 135° to the x-axis, characterizes the spatial correlation structure. The model contains eight injectors and eight producers, as shown in Fig. 1. The background of Fig. 1 shows the log of the x-direction transmissibility (transmissibility is in units of md-ft) for one of the realizations. Transmissibility is an interface quantity, and in this and subsequent figures, the blocks displayed are centered on the interfaces. The color of the block indicates the magnitude of log-transmissibility for that connection. Because transmissibility is an interface quantity, there are 44 blocks in the x-direction in Fig. 1 rather than 45 as in the geological model.

All wells are assumed to be stimulated to the same level, so all well indices are set to 3000 md-ft (this constant well index specification can also be viewed as conditioning to well permeability data). The reservoir is initially at a pressure of 5800 psi and saturated with oil. The production wells operate at a bottomhole pressure (BHP) of 5200 psi and the injection wells operate at a BHP of 6000 psi. The simulations are run for 800 days.

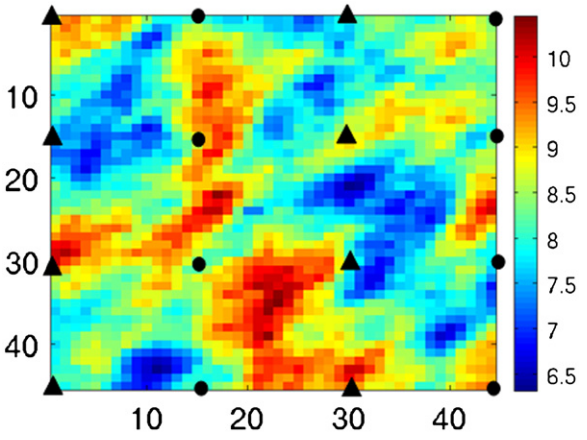

Fig. 1. Well locations for Model 1 (circles indicate producers and triangles indicate injectors). Background displays log-transmissibility for one realization.

3.2. Error definition

To assess the performance of the TPWL model, it is useful to compute errors (relative to full-order simulation) for quantities of interest. Here, errors are quantified in terms of the mismatch of the production rates (for both oil and water) and water injection rates between the full-order solution (Q_{full}) and TPWL simulations (Q_{tpwl}). For example, for oil production rate, the error for well j (E_o^j) is calculated as

$$E_o^j = \frac{\int_0^T |Q_{o,full}^j - Q_{o,tpwl}^j| \, dt}{\int_0^T Q_{o,full}^j}, \tag{18}$$

where subscript o designates oil, T is the total simulation time and the integration is accomplished using the trapezoidal rule over the time series of the flow rate data. The overall average error of the oil production rates, designated E_o, is computing by averaging E_o^j over all production wells

$$E_o = \frac{1}{n_{pw}} \sum_{j=1}^{n_{pw}} E_o^j, \tag{19}$$

where n_{pw} is the total number of production wells. Similar expressions are used to compute average water production error and water injection error (E_w and E_{inj}, respectively).

3.3. TPWL results

We now evaluate the accuracy of the TPWL model for predicting flow results on new realizations. We performed a sequence of TPWL evaluations using different levels of permeability variability, as quantified by the standard deviation of log k, denoted as $\sigma_{\log k}$. Values of $\sigma_{\log k}$ of 0, 0.25, 0.5, 0.75 and 1 were considered.

For each value of $\sigma_{\log k}$, 500 geological realizations were generated through application of SGEMS. These realizations were then used to construct the K–L representation given by Eq. (11) (using $l_m = 300$). From this K–L representation, 50 random realizations were generated by randomly sampling each component of ξ from a standard normal distribution. Training simulation runs were then performed using these realizations, and the TPWL model was constructed. Next, 50 new realizations were generated, again using the K–L representation, and (test) simulations were performed to evaluate the TPWL model performance. We emphasize that the new realizations are sampled from the same distribution used in the TPWL model construction (i.e., they are characterized by the same spatial correlation structure and $\sigma_{\log k}$).

The TPWL model constructed using the 50 training simulations (for a given value of $\sigma_{\log k}$) resulted in about 2500 saved states and matrices. The resulting $\mathbf{\Phi}$ matrix is characterized by $l_p = l_s = 200$ and $l_{lr} = 50$, where the latter denotes the number of local-resolution grid blocks (i.e., grid blocks retained at high resolution and not projected into the reduced space). Thus $\mathbf{\Phi}$ contains $l_p + l_s + 2l_{lr} = 500$ columns. TPWL simulations were then performed, and the average errors for oil and water production rates and water injection rates for the 50 test cases were calculated.

Fig. 2 shows the error in oil production rate, water production rate and water injection rate versus $\sigma_{\log k}$ for the TPWL model generated using 50 training runs (solid curves). All three errors increase linearly until $\sigma_{\log k} = 0.75$, beyond which there is a sharper increase. Water production rate displays more error than the other two quantities, possibly because of the differences in breakthrough times between the training and test runs.

Also shown in Fig. 2 are analogous results in which the TPWL model was constructed by generating and simulating only 20 (training) realizations. These results are clearly less accurate than those using 50 training simulations, as would be expected. Comparison of the two sets of results allows us to quantify the trade-off between TPWL model accuracy and the computational effort required to construct the TPWL model.

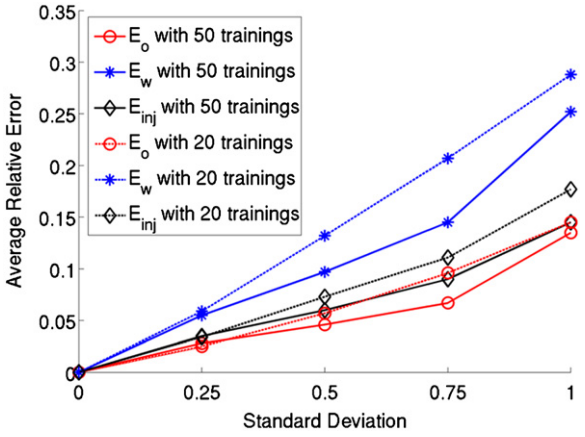

Fig. 2. Average TPWL errors for test cases as a function of $\sigma_{\log k}$ for Model 1. Results are for TPWL models constructed using 50 training runs (solid curves) and 20 training runs (dashed curves).

Figs. 3–5 display the oil production rate, water production rate and water injection rate, respectively, for a test case (using the TPWL model constructed from 50 training runs) with $\sigma_{\log k} = 0.5$. The errors (E_o, E_w, E_{inj}) for this case are 4.2%, 8.7% and 4.9%. These errors are close to the average errors for $\sigma_{\log k} = 0.5$ test cases (as is evident from Fig. 2), so these results can be considered to be representative. Results are shown only for four producers and four injectors, but these are representative of the results for the other eight wells. In the figures, the solid lines (which are red in the online version) show the reference solution from the full-order simulation (GPRS), the lines with circles (blue in the online version) show the solution provided by the TPWL model, and the dashed lines are the training simulation results about which the TPWL model is linearized.

It is evident that the TPWL results for the test case (lines with circles) are in reasonable agreement with the full-order results for production and injection rates, even though the training and test solutions differ considerably. This indicates that the TPWL model is able to capture the essential features of the test runs through linearization around (nearby) training runs. This feature will be important in our use of TPWL for history matching with ensemble methods such as EnKF. It should be noted, however, that the TPWL results do display some error, and that this error will clearly increase with increasing $\sigma_{\log k}$.

In terms of computational effort, the runtime for the full-order models for this case was about 30 s on an Opteron dual-core CPU. The TPWL models, by contrast, required less than 1 s. However, the construction of TPWL model requires simulating 50 training cases, plus additional overhead, which approximately doubles the cost. Therefore, it would not make sense to construct the TPWL model unless it is to be used for a large number of simulations. Because many simulations are required in history matching applications, the TPWL model should be applicable in this context. The use of TPWL in conjunction with an EnKF data assimilation procedure is explored in the following sections.

4. Use of TPWL for EnKF-based history matching

In this section we briefly describe the ensemble Kalman filter (EnKF) method, discuss some of its limitations, and show how we use TPWL to enhance the performance of EnKF.

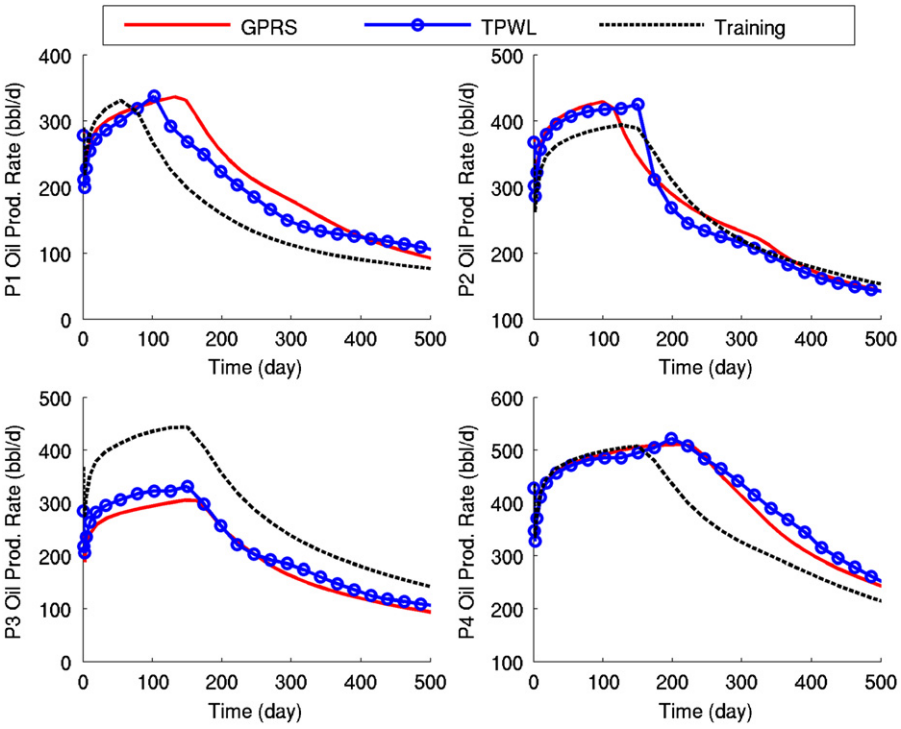

Fig. 3. Oil production rate for a representative test case with $\sigma_{\log k} = 0.5$ (Model 1).

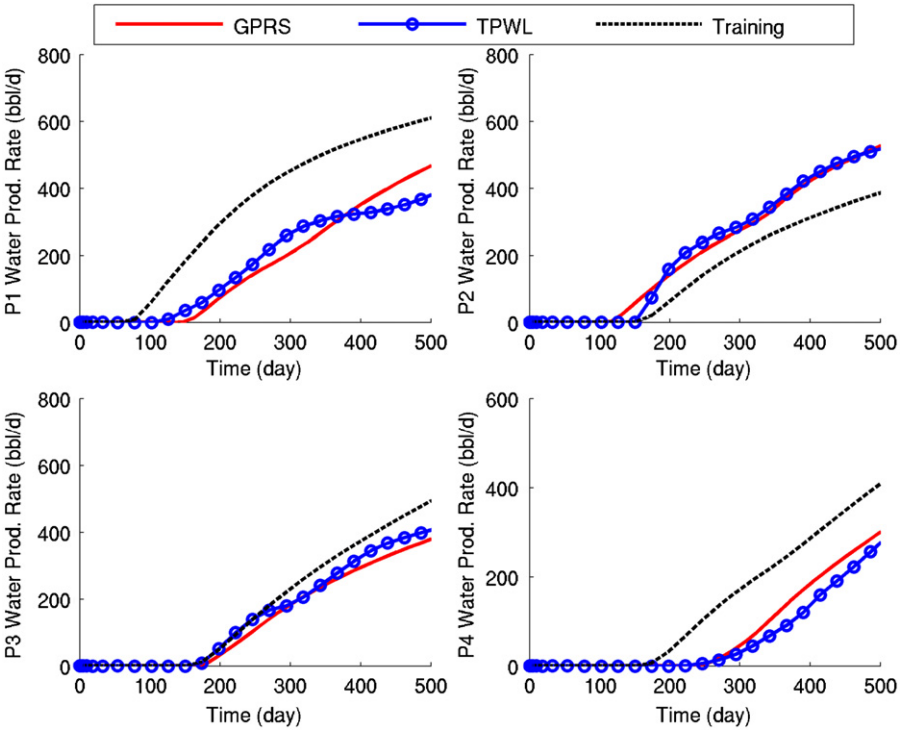

Fig. 4. Water production rate for a representative test case with $\sigma_{\log k} = 0.5$ (Model 1).

4.1. Basic ensemble Kalman filter method

EnKF was first proposed by Evensen (1994) and has since been applied by many researchers for history matching oil reservoirs. A recent review by Aanonsen et al. (2009) summarizes the history, current state, and some of the challenges associated with EnKF for reservoir engineering. These references should be consulted for details on EnKF. For the sake of completeness, an abbreviated description of the method is included here.

EnKF starts from an ensemble of initial reservoir models along with a priori geostatistical assumptions. The method then seeks to drive ensemble members to match production data at every assimilation step by correcting the models based on data mismatch and on the statistical correlation between members.

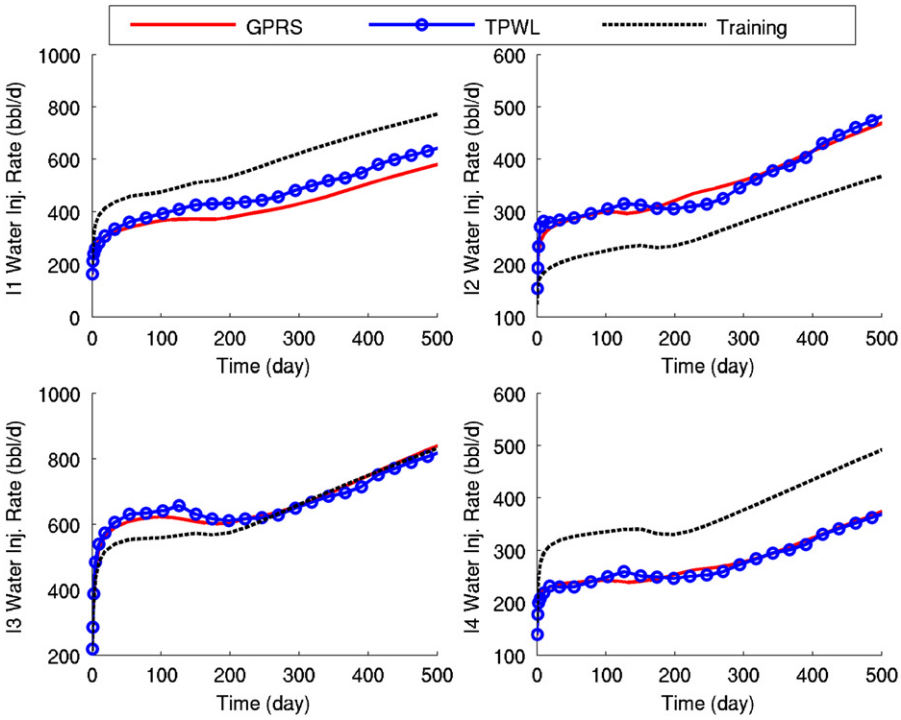

Fig. 5. Water injection rate for a representative test case with $\sigma_{\log k} = 0.5$ (Model 1).

We denote an ensemble of N_e state vectors at data assimilation step i as

$$\mathbf{Y}_i = [\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, \ldots, \mathbf{y}_{i,N_e}]. \tag{20}$$

The 'state vector' $\mathbf{y}$ in EnKF usually consists of the static model parameters $\mathbf{m}$ (permeability or transmissibility, porosity), the dynamic states $\mathbf{x}$ (pressure and saturation), and the predicted data $\mathbf{d}$ (well production rates, bottomhole pressures)

$$\mathbf{y}_{i,j} = \begin{bmatrix} \mathbf{m} \\ \mathbf{x} \\ \mathbf{d} \end{bmatrix}_{i,j}, \quad j = 1, \ldots, N_e. \tag{21}$$

The predicted data $\mathbf{d}$ are computed from the model variables $\mathbf{m}$ and $\mathbf{x}$. In history matching we attempt to match $\mathbf{d}$ with the measurement (or observed) data $\mathbf{d}_{obs}$ by modifying $\mathbf{m}$. Because the predicted data are included as part of the state vector, the relation between the state vector and the data can be viewed as linear. Therefore we have

$$\mathbf{d}_{i,j} = \mathbf{H}\mathbf{y}_{i,j}, \tag{22}$$

where $\mathbf{H}$ is an extraction matrix that selects the predicted data from the state vector.

EnKF updates reservoir models incrementally and assimilates data sequentially. As an ensemble-based method, the final result of EnKF is an ensemble of history matched reservoir models rather than a single model. Using the representation above, the EnKF procedure can be summarized as follows:

1. Create an ensemble of state vectors using prior geologic knowledge and initial reservoir conditions
$$\mathbf{Y}_0 = [\mathbf{y}_{0,1}, \mathbf{y}_{0,2}, \ldots, \mathbf{y}_{0,N_e}]. \tag{23}$$
2. *Forecast step*: Advance the state vectors to the next assimilation point using a simulator.
3. Compute the Kalman gain $\mathbf{K}_{e,i}$ at data assimilation step i as follows (Wen and Chen, 2006):
$$\mathbf{K}_{e,i} = \mathbf{C}^f_{\mathbf{Y},i}\mathbf{H}^T(\mathbf{H}\mathbf{C}^f_{\mathbf{Y},i}\mathbf{H}^T + \mathbf{C}_{D,i})^{-1}, \tag{24}$$
where $\mathbf{C}^f_{\mathbf{Y},i}$ is the covariance matrix of $\mathbf{Y}$ at step i, the superscript f denotes quantities calculated from the forecast step, and $\mathbf{C}_{D,i}$ designates the covariance of the error in observed data $\mathbf{d}_{obs}$ at step i.
4. *Assimilation step*: Update each member in the ensemble according to
$$\mathbf{y}^a_{i+1,j} = \mathbf{y}^f_{i,j} + \mathbf{K}_{e,i}(\mathbf{d}_{i,j} - (\mathbf{d}_{obs})_{i,j}), \tag{25}$$
where the superscript a denotes that the quantity is calculated in the assimilation step.
5. Return to step 2 if there are additional data to assimilate.

EnKF does have several limitations, one of which is associated with the ensemble size. The technique is often observed to perform well when the ensemble size is sufficiently large, but a large ensemble size means that many simulations are required, which may not be practical for high-resolution models. However, when a small ensemble size is used, errors in the estimation of the covariance matrix, and thereby the Kalman gain, can be large, which results in unrealistic updates of the state vector. In such cases, after a few assimilation steps the variability can become so low that the entire ensemble collapses to an incorrect reservoir model (Lorenc, 2009). In addition, the number of degrees of freedom in EnKF is limited by the ensemble size. A small ensemble may not have enough degrees of freedom to represent the true solution accurately.

The most common approach for reducing the errors in covariance estimation due to a small ensemble is to apply distance-based localization. In this method, the covariance of spatial

variables is assumed to be dependent on the distance and is cut off (or tapered) when the distance is large (Houtekamer and Mitchell, 1998; Houtekamer et al., 2005). Although the method is straightforward to apply and can avoid ensemble collapse when an appropriate choice of tapering range is used, determining an optimal tapering function can be difficult. In addition, the optimal localization depends not only on the type of data to be assimilated but also on previous data assimilation steps (Chen and Oliver, 2010).

For some variables, distance-based localization may not be applicable, and localization methods that do not assume the distance dependence of covariance have also been suggested. In particular, Zhang and Oliver (2010) proposed a method using bootstrap sampling to estimate the reliability of the Kalman gain and subsequently remove unreliable terms. It is important to note, however, that most localization techniques modify the Kalman gain without considering its impact on the prior geostatistics, which may alter the geological correlation structure. Thus, in light of the issues noted here, it is clearly worth exploring alternatives to existing localization procedures.

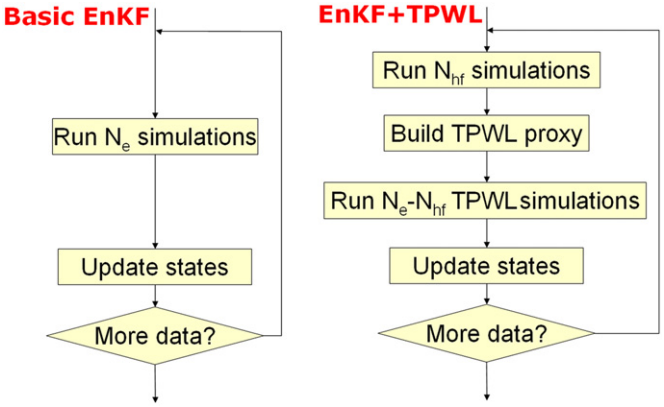

Fig. 6. Flowchart of basic EnKF and EnKF + TPWL methods.

4.2. Combining EnKF and TPWL for history matching

The TPWL representation has some features that render it well suited for use with EnKF. Most importantly, the runtime for TPWL models is on the order of one second, which is much less than the time required to simulate a typical high-fidelity model (though the computational requirements for TPWL overhead must also be considered). In addition, TPWL enables us to recover the full-order states from the reduced states (through use of Eqs. (5) and (11)), which is required for EnKF updates.

The way in which we incorporate TPWL into EnKF is illustrated in the flowchart shown in Fig. 6. As in standard EnKF, we first perform a forecast step. However, instead of simulating each member of the ensemble using a high-fidelity simulation, we use full-order simulations for only some of the members of the ensemble. After these simulations are performed, they are used as training runs for TPWL. Then, TPWL is applied to simulate the remaining members in the ensemble. Flow results from both the high-fidelity runs and the TPWL runs are assimilated in EnKF to provide the updated state vectors. This process is repeated until all data are assimilated.

Given an ensemble of N_e realizations, we need to select N_{hf} members of the ensemble for high-fidelity (GPRS) simulation (N_{hf} is analogous to N_ω used earlier). This is accomplished by performing clustering based on the distance definition (d_T^y) in Eq. (16). To accomplish this, we apply k-means clustering (Lloyd, 1982) at the beginning of each forecast step to cluster the N_e realizations into N_{hf} clusters. The realizations that are closest to the center of each cluster are used for full-order simulation. These realizations are expected to be reasonably well distributed over the parameter space. It will be of interest to explore other approaches for selecting the realizations for high-fidelity simulation, as well as methods to determine the appropriate value of N_{hf}.

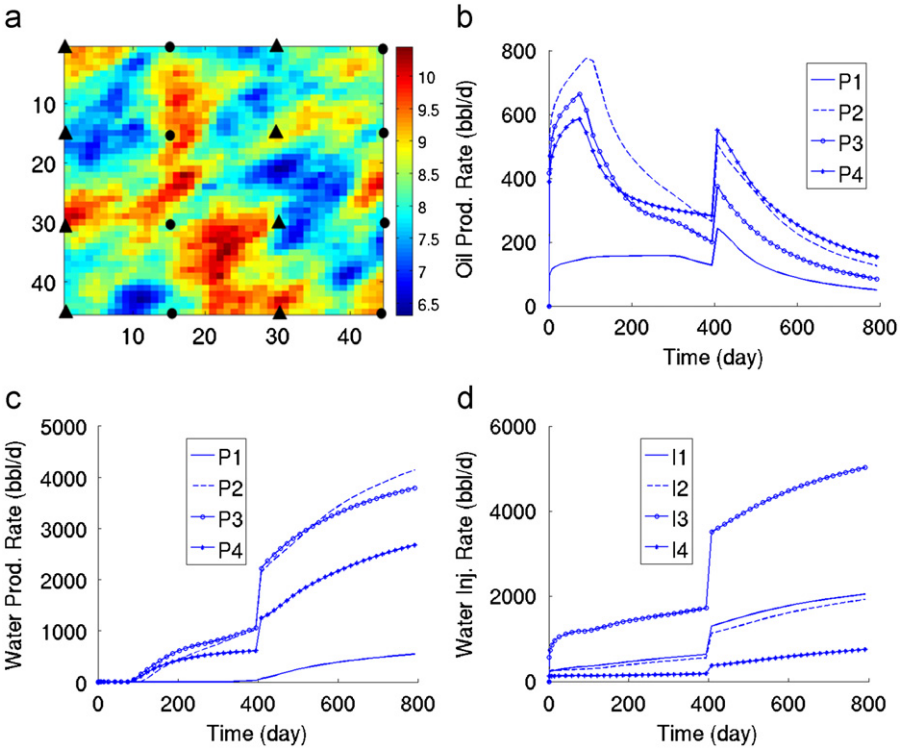

Fig. 7. Reference log-transmissibility field and well rate data. (a) Reference log-transmissibility field. (b) Q_o at four production wells. (c) Q_w at four production wells. (d) Q_{inj} at four injection wells.

5. History matching using TPWL-based EnKF

5.1. Model 1: problem set up

We consider a history matching problem involving Model 1, presented in Section 3.1. The initial ensemble of permeability realizations is generated using $\sigma_{\log k}=1$ (other geostatistical parameters are as described earlier). The uncertain parameters for the history matching are the 3960 transmissibility values. Other parameters (porosity of 0.3, irreducible water saturation of 0.2) are assumed to be known.

The well specifications are different from those used in Section 3.1. The reservoir is initially saturated with oil and is at a pressure of 5800 psi. The simulations are run for 800 days. The production wells operate at a BHP of 5600 psi from 0 to 400 days and at 5400 psi from 400 days to 800 days. The injection wells operate at a BHP of 6000 psi from 0 to 400 days and at 6200 psi from 400 days to 800 days.

Fig. 7(a) shows the 'true' transmissibility model, which was used to generate the production data. This transmissibility field is characterized by the Model 1 geostatistics, but it is not one of the initial realizations used in the EnKF procedure. The oil production rates (Q_o), water production rates (Q_w) and water injection rates (Q_{inj}) for the first four producers and injectors from this model are shown in Fig. 7(b)–(d), respectively. Of the total 800-day production period, the first 400 days are treated as the production history and the second 400 days are treated as the prediction period. During the production history we assume that Q_o, Q_w and Q_{inj} at each well are available to us every 50 days. There are thus a total of eight assimilation steps with 24 pieces of data at each assimilation step. Gaussian errors with zero mean and a standard deviation of 3 stb/day were added to the true production data to create an ensemble of noisy production data.

As in Section 3.3, in the TPWL model we use $l_p=l_s=200$ and $l_{lr}=50$ (thus $\mathbf{\Phi}$ contains 500 columns), and $l_m=300$. Around seven snapshots were collected from each training simulation for each forecast period. Therefore, there were about 350 snapshots for each 50 day interval.

Three different scenarios are considered. In the first scenario (referred to as HF200), the EnKF ensemble consists of 200 randomly sampled members which, as shown later, is large enough to avoid ensemble collapse. For this scenario all models were simulated at high fidelity (i.e., TPWL is not applied). Note that HF200 provides the reference against which the other two cases are compared. In the second scenario (HF50), the ensemble consists of the first 50 of the 200 models in HF200. Thus HF50 contains an ensemble of 50 randomly sampled members. All models were again simulated at high fidelity. In the third scenario (HF50+TPWL150), the EnKF ensemble is the same as in HF200, though only 50 of the models were simulated at high fidelity. The other 150 members were simulated using TPWL, with the TPWL models constructed as described earlier.

5.2. Model 1: history matching results

Fig. 8 shows both the history match and the prediction of oil production rate for the first production well for the three cases. These results were obtained by re-running the final updated ensemble for the entire simulation period. Shown in Fig. 8(a) are the results from the initial ensemble (solid lines, blue in the online version), with the reference (true) solution (dashed lines, red in the online version). It is clear that the variability in the initial ensemble is large. Fig. 8(b) shows the results from the 200 realizations after EnKF assimilation for HF200 for Producer 1. These results are representative of those for all of the producers. We see that, for the history matching period, the spread of the ensemble results is significantly reduced relative to that in Fig. 8(a), which demonstrates that the assimilation of production data improves the match between all ensemble members and the true solution significantly, as expected. Note also that the reference solution always falls within the ensemble results (during both the production and prediction periods).

Fig. 8(c) shows the results obtained for HF50. Because of the small ensemble size, the curves collapse to a very small range.

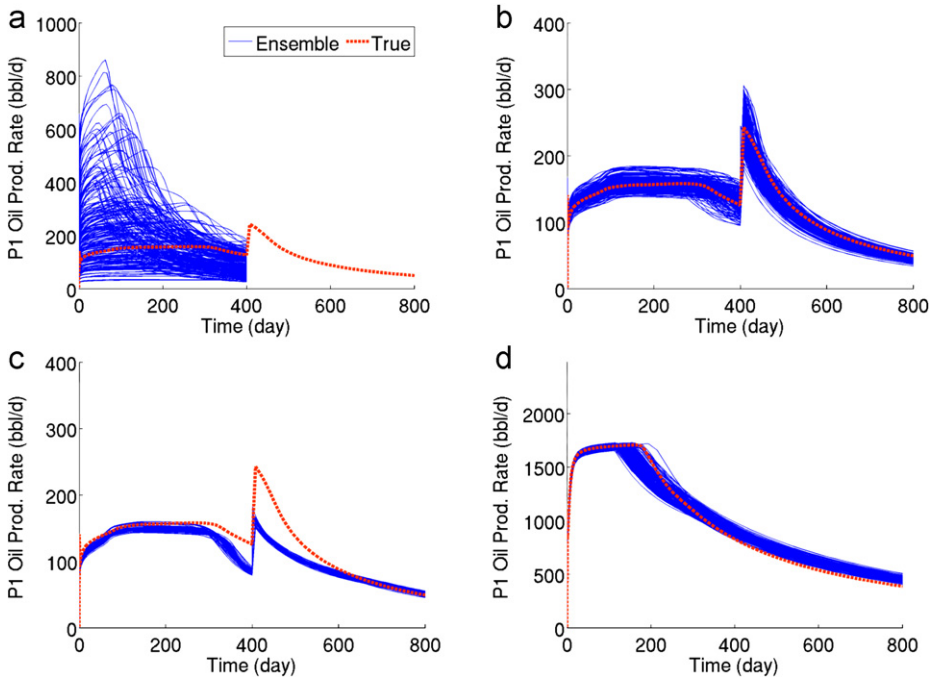

Fig. 8. Oil production rate for initial and final updated ensembles for Producer 1 in Model 1. (a) Initial. (b) HF200. (c) HF50. (d) HF50+TPWL150.

The reference solution does not fall within this range, which illustrates the degradation that can occur with EnKF when the size of the ensemble is too small. The results achieved using HF50+TPWL150 are shown in Fig. 8(d). It is evident that the general solution is very near that of the reference case (HF200), and of much better quality than HF50.

Figs. 9 and 10 show the water production rates and water injection rates for the three cases. For water production, clear improvement relative to the HF50 case is achieved through the use of HF50+TPWL150. For water injection, HF50+TPWL150 results in increased spread relative to the reference case which indicates lower history matching quality compared with the reference case. This is due to error introduced by the TPWL approximation. Nonetheless, the true solution always falls within the ensemble results, in contrast to the results for HF50.

We now quantify the error introduced by TPWL at each forecast step by computing E_o, E_w and E_{inj} for the TPWL runs (this requires us to run the same simulations using the full-order simulator). Fig. 11 displays average TPWL flow rate errors for each EnKF step. The errors for E_o and E_{inj} decrease as EnKF proceeds as the variance between realizations decreases. This decrease in error is to be expected because the test cases become 'closer' to the training runs as the differences between realizations decrease. The water production rate error increases between the first and second EnKF steps, but this is only because water breakthrough has not occurred at the first step. After the second step, this error also decreases. It is notable that, at the end of the EnKF process, the largest error is in water injection rate. This is consistent with the results in Fig. 10, where the spread in the results for the EnKF-TPWL method is greater than that for HF200, in contrast to the results for oil and water production rates.

Fig. 11 suggests that it may be useful to develop a TPWL-based EnKF procedure that uses more training simulations at earlier stages, when the variability of the ensemble is large, and fewer at the later stages. An approach along these lines could lead to improved accuracy and/or reduced computational requirements. It will be worthwhile to pursue such a procedure in future work.

Fig. 12 displays the mean log-transmissibility field (expectation) and the associated variance field computed from the initial ensemble and the final ensemble for the three scenarios. Before any data have been assimilated, the mean and variance are largely featureless. After EnKF assimilation for HF200, it is clear that the mean of the ensemble captures many of the characteristics of the reference field shown in Fig. 7(a). In addition, the variance is reduced in the vicinity of well locations. For HF50, because of ensemble collapse, the variance is reduced to nearly zero at all locations, meaning that all members of the ensemble are very similar. It is also clear that the mean of the ensemble in this scenario contains many extreme values and does not retain the smoothness and continuity of the reference field. Results for HF50+TPWL150 resemble the reference HF200 results and do not display the anomalies evident in the HF50 results. This further demonstrates the potential of TPWL-based EnKF.

Fig. 13 shows the mean and variance computed using a recently proposed localization method (Zhang and Oliver, 2010) with an ensemble of 50 realizations. It is evident that this localization technique does avoid ensemble collapse and provides much better results than HF50. However, the variance is still lower than that of the reference scenario and the mean field is not as smooth as in the reference case. By these measures, our method may offer some advantages over localization.

5.3. Model 2: problem set up

This example, designated Model 2, involves a three-dimensional system defined on a $30 \times 30 \times 10$ grid with cells of size $50\ \text{m} \times 50\ \text{m} \times 50\ \text{m}$. The realizations were generated using sequential Gaussian simulation in SGEMS (Remy et al., 2008). The $\log k$ field is characterized by a Gaussian distribution with mean of 5.0 and $\sigma_{\log k} = 1$. A spherical variogram, with ranges of 1000 m, 500 m and 100 m, oriented at 45° to the x-axis, 135° to the x-axis, and vertically, was used to generate the realizations. The parameters to be determined using EnKF are the 25,500 transmissibility values. These transmissibilities are represented

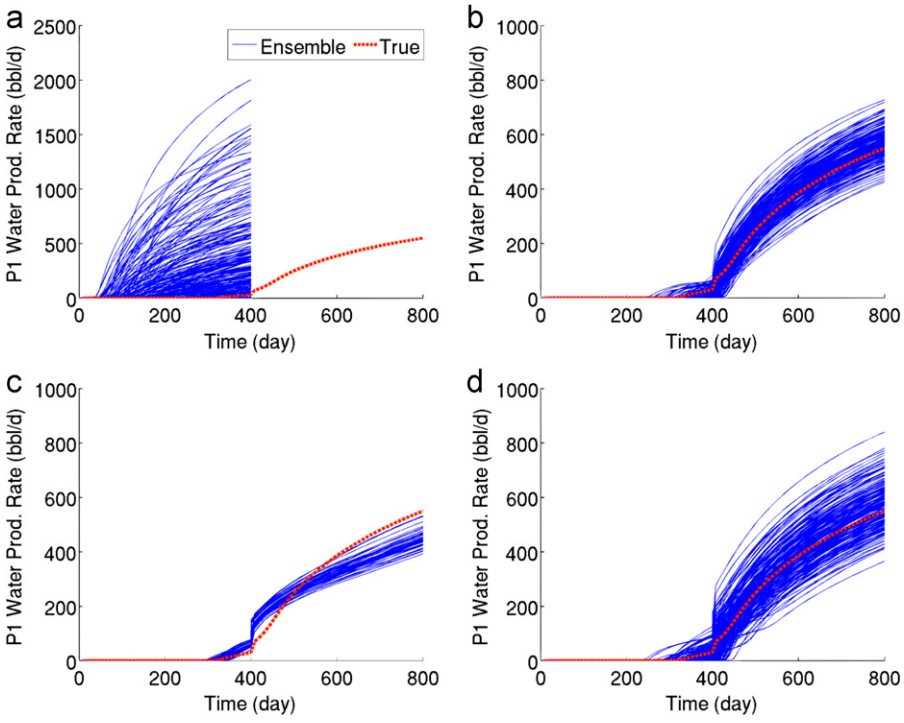

Fig. 9. Water production rate for initial and final updated ensembles for Producer 1 in Model 1. (a) Initial. (b) HF200. (c) HF50. (d) HF50+TPWL150.

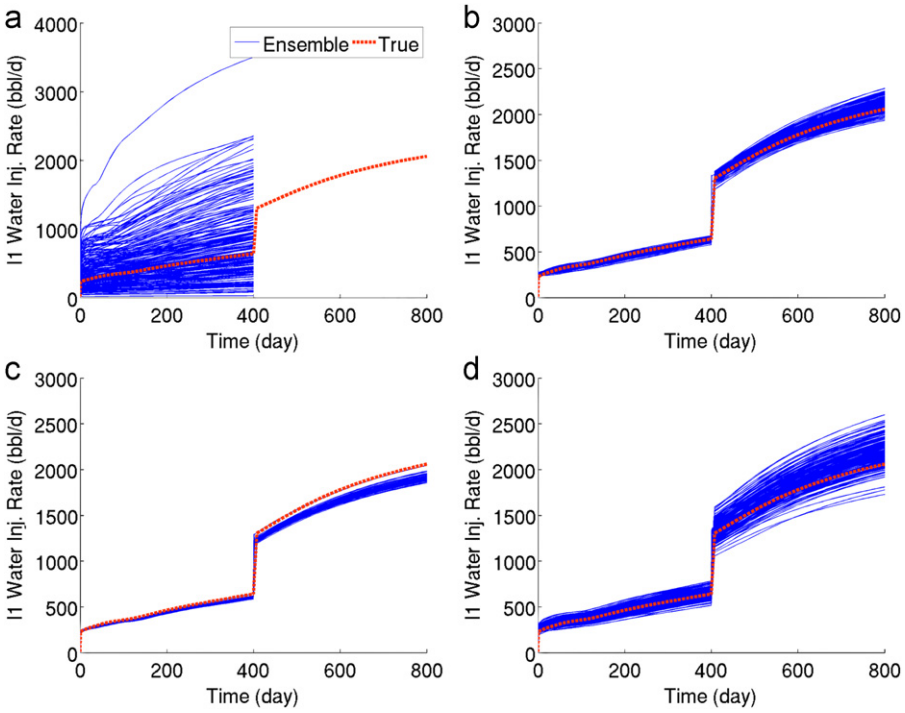

Fig. 10. Water injection rate for initial and final updated ensembles for Injector 1 in Model 1. (a) Initial. (b) HF200. (c) HF50. (d) HF50+TPWL150.

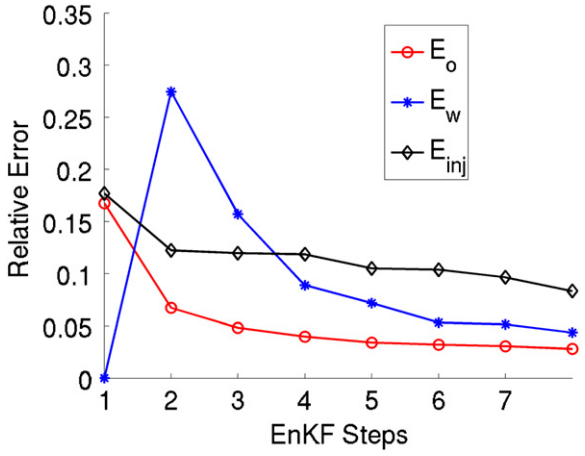

Fig. 11. Average TPWL error at each EnKF step.

using 300 parameters in the K–L expansion (l_m=300). Other parameters (porosity of 0.3, initial water saturation of 0.2) are assumed known. The 18,000 state variables are reduced to a total of 500 variables, as in the previous example.

There are eight injectors and eight producers arranged in a five-spot pattern. One log-transmissibility realization, along with the well locations, is shown in Fig. 14. All wells are assumed to be stimulated to the same level, and their well indices are set to be 4000 md-ft. The reservoir is initially saturated with oil and it is at a pressure of 5800 psi. The simulations are run for 800 days. The production wells operate at a BHP of 5600 psi and the injection wells operate at 6000 psi. Of the total 800-day production period, the first 600 days are treated as the production history and the second 200 days are treated as the prediction period. Data for Q_o and Q_w at producers, and Q_{inj} at injectors, are provided every 30 days. This gives a total of 20 assimilation steps with 24 data measurements at each step. Gaussian errors with zero mean and a standard deviation of 3 stb/day are added to the true production data to create an ensemble of noisy data.

We again consider three different scenarios: the first (reference) case, denoted HF200, includes 200 high-fidelity members; the second scenario, HF50, contains 50 high-fidelity members; and the third scenario, HF50+TPWL150, entails 50 high fidelity members and 150 TPWL members.

5.4. Model 2: history matching results

Figs. 15–17 present the history matching results for oil production rate, water production rate and water injection rate, respectively, for the first producer and injector for the initial ensemble and for the three EnKF scenarios. These results are representative of those for the other wells. The variability in the initial ensemble is reduced in all three scenarios. We again observe reasonable agreement between the HF200 and HF50+TPWL150 scenarios, indicating the viability of our approach. There is some increased spread in the EnKF results using HF50+TPWL150 relative to those for HF200, but the inaccuracy and ensemble collapse evident in the HF50 results are clearly avoided.

In terms of computational effort, the runtime for the full-order models for this case was about 100 s on an Opteron dual-core CPU. The TPWL models, by contrast, required less than 1 s to run. The overall speedup is much less than this ratio, however, because the construction of the TPWL model requires simulating 50 training cases plus additional overhead. Larger overall speed-ups could be achieved by using fewer training simulations at later stages of the EnKF (when the variability of the ensemble is relatively small), or by using more total realizations in the ensemble (but still taking only 50 training runs). In either case, a larger fraction of the models could be simulated using TPWL, which would reduce computational requirements.

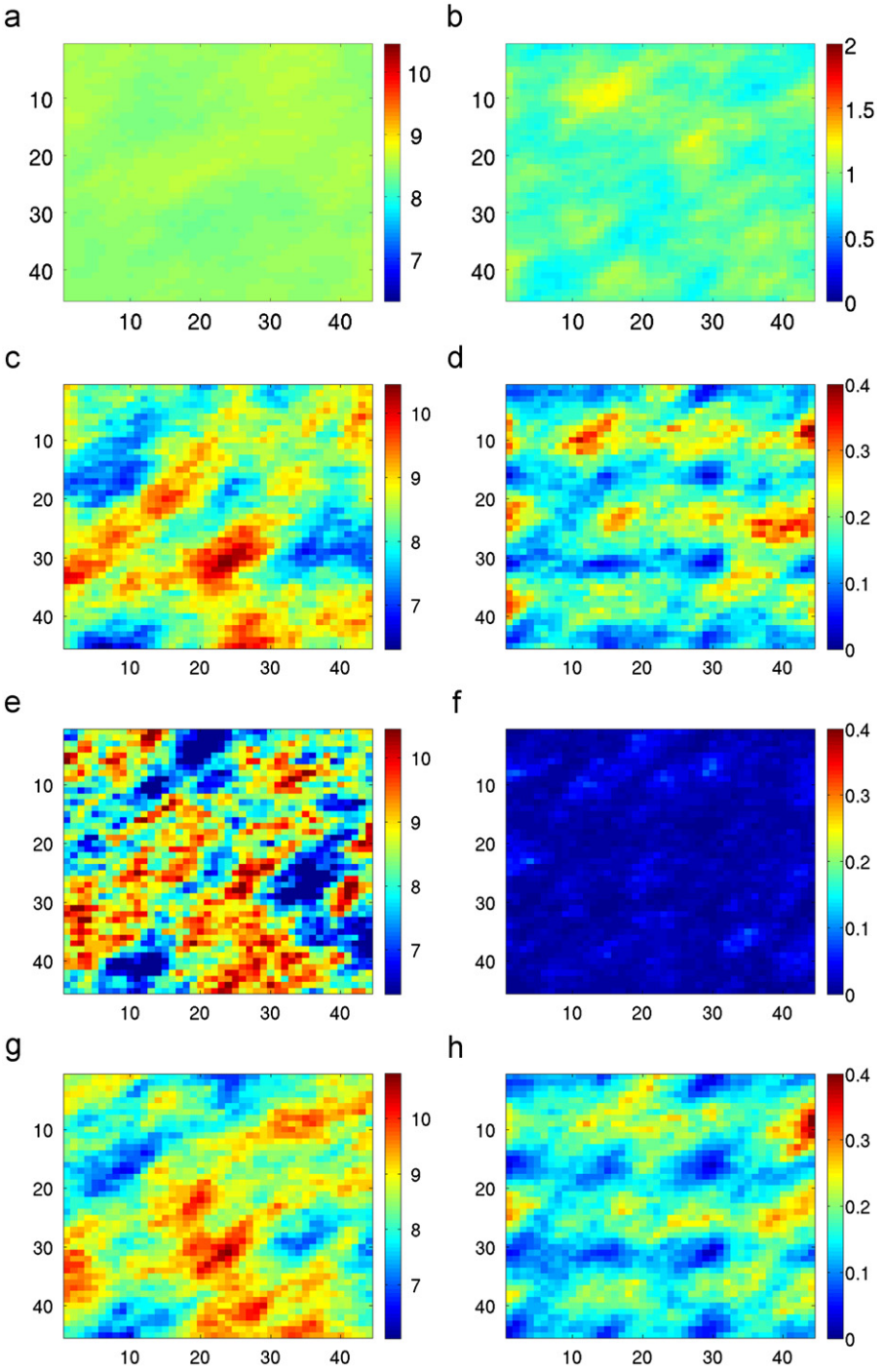

Fig. 12. Mean and variance of the initial ensemble and final ensembles for the three scenarios (Model 1). (a) Mean: initial. (b) Variance: initial. (c) Mean: HF200. (d) Variance: HF200. (e) Mean: HF50. (f) Variance: HF50. (g) Mean: HF50+TPWL150. (h) Variance: HF50+TPWL150.

6. Concluding remarks

In this paper, we introduced a reduced-order modeling procedure for data assimilation based on trajectory piecewise linearization (TPWL). By linearizing around previously simulated training runs and performing model-order reduction using proper orthogonal decomposition, the TPWL description is able to provide approximate simulation results for new geological models. The geological model can be described in terms of the log-transmissibility field, and this field can in turn be represented using a Karhunen–Loève expansion. The overall procedure thus provides a very concise (low-order) description of both the dynamic states and the geological model parameters. This enables simulations for new geological models to be performed very efficiently. The TPWL model construction, however, does require multiple training simulations and additional overhead computations. Thus, this approach is only appropriate when many models are to be simulated, as is the case with ensemble-based data assimilation methods.

In this work the TPWL surrogate model was incorporated into an ensemble Kalman filter (EnKF) history-matching procedure. The combined method was tested for both two- and three-dimensional systems, with only 1/4 of the realizations in the ensemble simulated at high fidelity (full order). For both systems, the new procedure was

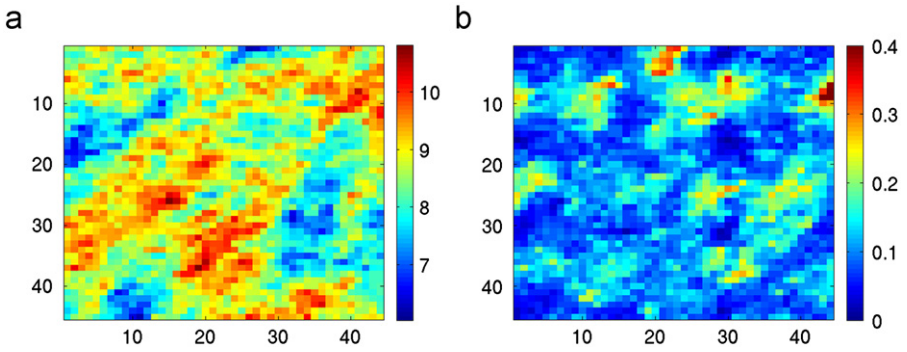

Fig. 13. Mean and variance of the final ensemble using localization (Model 1). (a) Mean. (b) Variance.

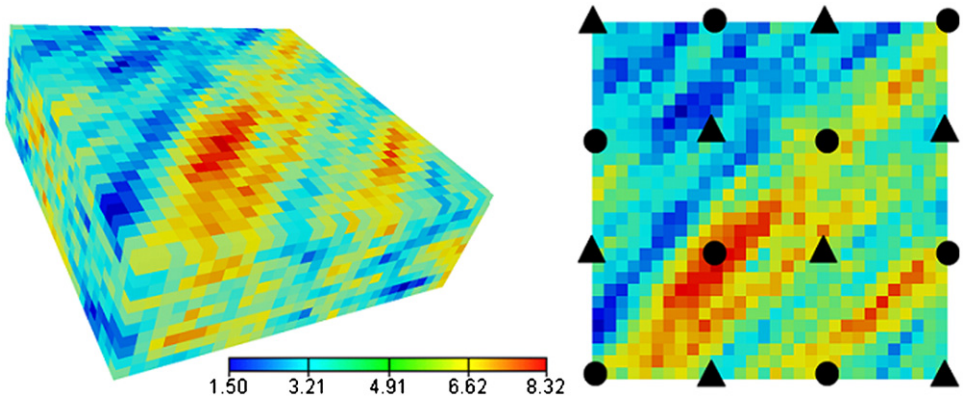

Fig. 14. Log-transmissibility (for one realization) and well locations for Model 2. Circles indicate producers and triangles indicate injectors.

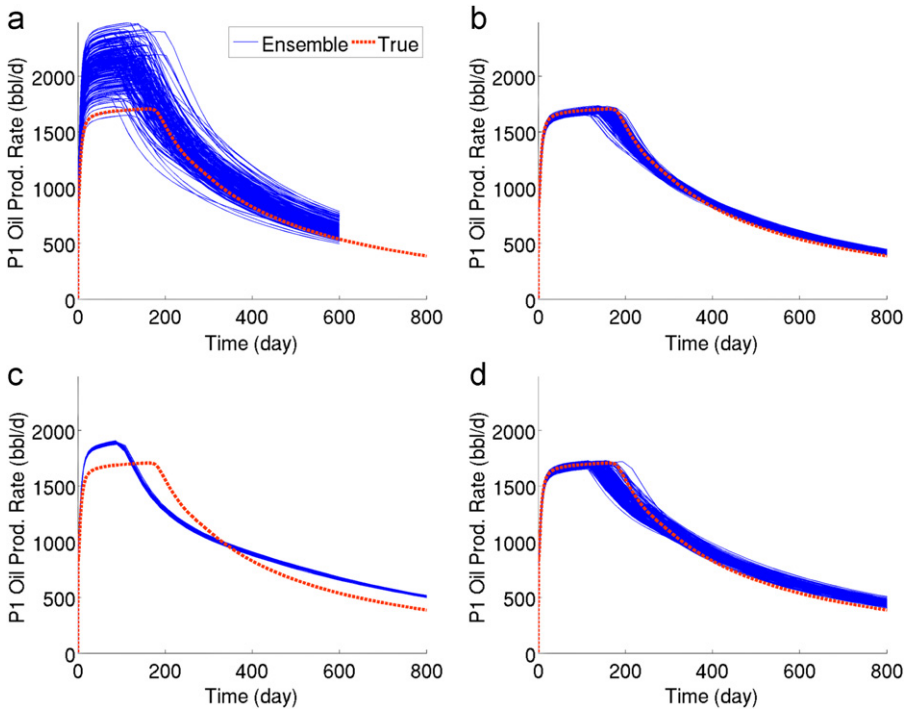

Fig. 15. Oil production rate for initial and final updated ensembles for Producer 1 in Model 2. (a) Initial. (b) HF200. (c) HF50. (d) HF50+TPWL150.

shown to provide results in reasonably close agreement with reference scenarios in which all simulations were performed at high fidelity. Significantly, results using the proposed method were much better than those using EnKF with a small ensemble. The speedups obtained for the EnKF example presented here are, however, still modest. Specifically, even though only 1/4 of the models (50 out of

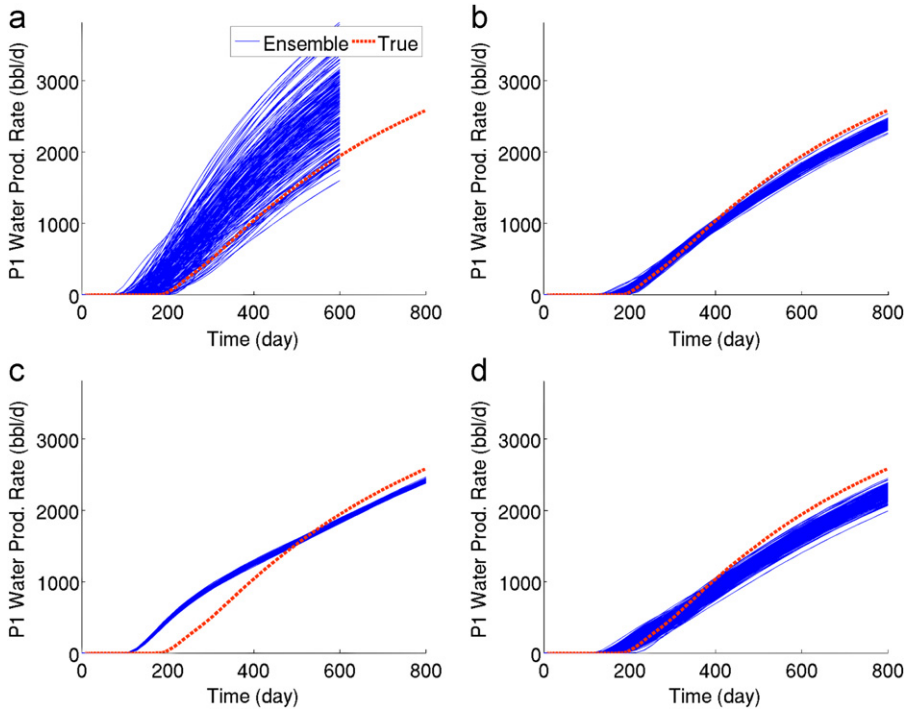

Fig. 16. Water production rate for initial and final updated ensembles for Producer 1 in Model 2. (a) Initial. (b) HF200. (c) HF50. (d) HF50+TPWL150.

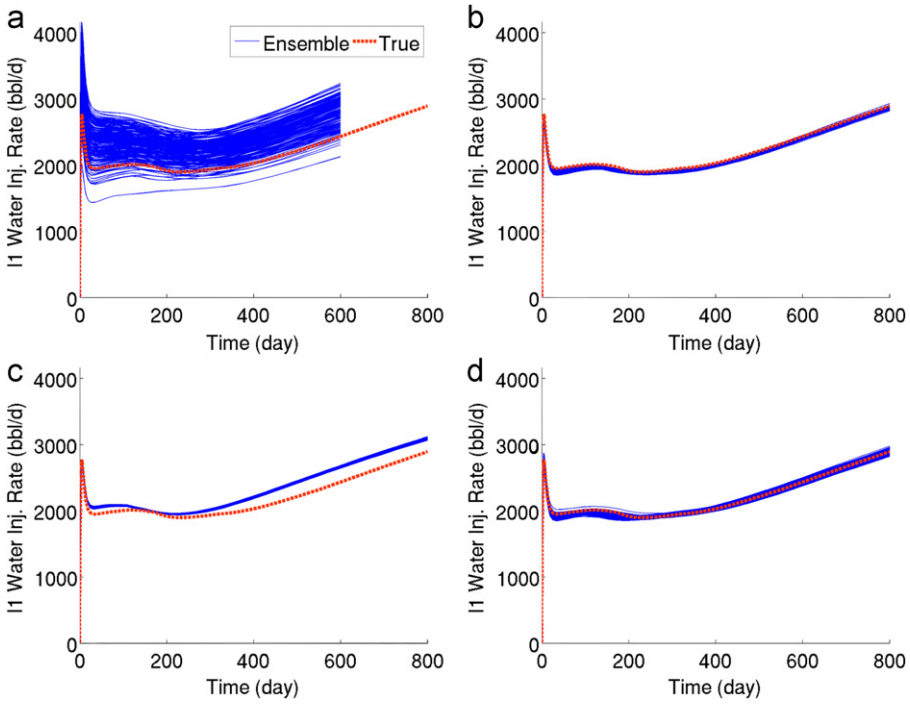

Fig. 17. Water injection rate for initial and final updated ensembles for Injector 1 in Model 2. (a) Initial. (b) HF200. (c) HF50. (d) HF50+TPWL150.

200) were simulated at high fidelity in the HF50+TPWL150 scenarios, the speedups achieved were closer to a factor of two (rather than a factor of 4) due to the TPWL overhead requirements. Because our current code is a Matlab implementation, we expect that the overhead computations could be accelerated, which would lead to somewhat better speedups.

There are, in addition, several ways in which more substantial speedups and/or improvements in accuracy could be achieved, and these should be investigated in future work. As suggested earlier, the number of high-fidelity simulations could be optimized (adaptively), which would lead to more full-order simulations when the variability of the ensemble is large and fewer as the variability decreases at later EnKF stages. The selection of which realizations to use for high-fidelity simulation should also be investigated. Our approach here applies a clustering procedure based on a particular metric, but other treatments for this

determination, and other metrics, should be considered. The overall method is also expected to benefit, in terms of speedup, if larger numbers of realizations are included in the ensemble. The point selection strategy has significant impact on TPWL performance, and alternate treatments should be investigated.

It may also be possible to enhance the accuracy of the underlying TPWL representation, and progress along these lines could lead to benefits in application areas beyond data assimilation. The linear approximations used in TPWL could conceivably be improved by introducing semi-analytical estimates for key higher-order effects into Eq. (13). In addition, it will be of interest to explore approaches other than Galerkin projection, such as least-squares, for reducing the number of equations to be solved in TPWL. The optimal use of local resolution in TPWL should also be investigated.

The K–L representation applied here is best suited for use with Gaussian fields. For non-Gaussian fields, the application of TPWL with geological models represented in terms of kernel principal component analysis (Sarma et al., 2008) should be investigated. Finally, it will be useful to evaluate the performance of TPWL for other history matching techniques, including both gradient-based and stochastic procedures.

Acknowledgments

We are grateful to the industrial affiliates of the Stanford University Reservoir Simulation Research (SUPRI-B) and Smart Fields Consortia for partial funding of this work. We would also like to thank the Society of Petroleum Engineers for allowing us to include some of the content presented in He et al. (2011b).

References

Aanonsen, S.I., Oliver, D.S., Reynolds, A.C., Vallès, B., 2009. Ensemble Kalman filter in reservoir engineering: a review. SPE Journal 14 (3), 393–412.

Astrid, P., Verhoeven, A., 2006. Application of least squares MPE technique in the reduced order modeling of electrical circuits. In: 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, pp. 1980–1986.

Berkooz, G., Titi, E.Z., 1993. Galerkin projections and the proper orthogonal decomposition for equivariant equations. Physics Letters A 174 (1–2), 94–102.

Cao, H., 2002. Development of Techniques for General Purpose Simulators. Ph.D. Thesis. Stanford University.

Cardoso, M.A., Durlofsky, L.J., 2010. Linearized reduced-order models for subsurface flow simulation. Journal of Computational Physics 229 (3), 681–700.

Cardoso, M.A., Durlofsky, L.J., Sarma, P., 2009. Development and application of reduced-order modeling procedures for subsurface flow simulation. International Journal for Numerical Methods in Engineering 77 (9), 1322–1350.

Carlberg, K., Bou-Mosleh, C., Farhat, C., 2011. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. International Journal for Numerical Methods in Engineering 86 (2), 155–181.

Chen, Y., Oliver, D.S., 2010. Cross-covariances and localization for EnKF in multiphase flow data assimilation. Computational Geosciences 14, 579–601.

Evensen, G., 1994. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. Journal of Geophysical Research 99 (C5), 10143–10162.

Gratton, D., Willcox, K., 2004. Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics. In: 34th AIAA Fluid Dynamics Conference and Exhibit, Portland, OR.

Gu, Y., Oliver, D.S., 2005. History matching of the PUNQ-S3 reservoir model using the ensemble Kalman filter. SPE Journal 10, 217–224.

He, J., Sætrom, J., Durlofsky, L.J., 2011a. Enhanced linearized reduced-order models for subsurface flow simulation. Journal of Computational Physics 230, 8313–8341.

He, J., Sarma, P., Durlofsky, L.J., 2011b. Use of reduced-order models for improved data assimilation within an EnKF context. In: SPE Reservoir Simulation Symposium, SPE 141967, The Woodlands, TX.

Houtekamer, P.L., Mitchell, H.L., 1998. Data assimilation using an ensemble Kalman filter technique. Monthly Weather Review 126 (3), 796–811.

Houtekamer, P.L., Mitchell, H.L., Pellerin, G., Buehner, M., Charron, M., Spacek, L., Hansen, B., 2005. Atmospheric data assimilation with ensemble Kalman filter: results with real observations. Monthly Weather Review 133 (3), 604–620.

Jolliffe, I.T., 2002. Principal Component Analysis, second ed. Springer, New York.

Lloyd, S.P., 1982. Least square quantization in PCM. IEEE Transactions on Information Theory 28 (2), 129–137.

Lorenc, A.C., 2009. The potential of the ensemble Kalman filter for NWP—a comparison with 4d-var. Quarterly Journal of the Royal Meteorological Society 14 (3), 393–412.

Nævdal, G., Brouwer, D.R., Jansen, J.D., 2006. Waterflooding using closed-loop control. Computational Geosciences 10 (1), 37–60.

Nævdal, G., Johnsen, L.M., Aanonsen, S.I., Vefring, E.H., 2005. Reservoir monitoring and continuous model updating using ensemble Kalman filter. SPE Journal 10 (1), 66–74.

Nævdal, G., Mannseth, T., Vefring, E.H., 2002. Near-well reservoir monitoring using ensemble Kalman filter. In: SPE/DOE Improved Oil Recovery Symposium, SPE 75235, Tulsa, OK.

Oliver, D.S., 1996. Multiple realizations of the permeability field from well test data. SPE Journal 1, 145–154.

Park, K., 2011. Modeling Uncertainty in Metric Space. Ph.D. Thesis. Stanford University.

Remy, N., Boucher, A., Wu, J., 2008. Applied Geostatistics with SGeMS: A User's Guide. Cambridge University Press.

Rewienski, M., White, J., 2003. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 22 (2), 155–170.

Sarma, P., Durlofsky, L.J., Aziz, K., 2008. Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics. Mathematical Geosciences 40 (1), 3–32.

Sarma, P., Durlofsky, L.J., Aziz, K., Chen, W.H., 2006. Efficient real-time reservoir management using adjoint-based optimal control and model updating. Computational Geosciences 10, 3–36.

van Doren, J.F.M., Markovinović, R., Jansen, J.D., 2006. Reduced-order optimal control of water flooding using proper orthogonal decomposition. Computational Geosciences 10, 137–158.

Vasilyev, D., Rewienski, M., White, J., 2006. Macromodel generation for BioMEMS components using a stabilized balanced truncation plus trajectory piecewise-linear approach. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems 25 (2), 285–293.

Wen, X.H., Chen, W.H., 2006. Real-time reservoir model updating using ensemble Kalman filter with confirming option. SPE Journal 11 (4), 431–442.

Yang, Y.J., Shen, K.Y., 2005. Nonlinear heat-transfer macromodeling for MEMS thermal devices. Journal of Micromechanics and Microengineering 15 (2), 408–418.

Zhang, Y., Oliver, D.S., 2010. Improving the ensemble estimate of the Kalman gain by bootstrap sampling. Mathematical Geosciences 42, 327–345.