Delft University of Technology

Non-intrusive subdomain POD-TPWL for reservoir history matching

Xiao, Cong; Leeuwenburgh, Olwijn; Lin, Hai Xiang; Heemink, Arnold

CrossMark

Non-intrusive subdomain POD-TPWL for reservoir history matching

Cong Xiao[1] · Olwijn Leeuwenburgh[2,3] · Hai Xiang Lin[1] · Arnold Heemink[1]

Abstract

This paper presents a non-intrusive subdomain POD-TPWL (SD POD-TPWL) for reservoir history matching through integrating domain decomposition (DD), proper orthogonal decomposition (POD), radial basis function (RBF) interpolation, and the trajectory piecewise linearization (TPWL). It is an efficient approach for model reduction and linearization of general non-linear time-dependent dynamical systems without accessing to the legacy source code. In the subdomain POD-TPWL algorithm, firstly, a sequence of snapshots over the entire computational domain is saved and then partitioned into subdomains. From the local sequence of snapshots over each subdomain, a number of local basis vectors is formed using POD, and then the RBF interpolation is used to estimate the derivative matrices for each subdomain. Finally, those derivative matrices are substituted into a POD-TPWL algorithm to form a reduced-order linear model in each subdomain. This reduced-order linear model makes the implementation of the adjoint easy and results in an efficient adjoint-based parameter estimation procedure. Comparisons with the classic finite-difference-based history matching show that our proposed subdomain POD-TPWL approach is obtaining comparable results. The number of full-order model simulations required is roughly 2–3 times the number of uncertain parameters. Using different background parameter realizations, our approach efficiently generates an ensemble of calibrated models without additional full-order model simulations.

Keywords Data assimilation · Reduced-order modeling · Model linearization · Domain decomposition

Abbreviations

POD   Proper orthogonal decomposition
PCA   Principal component analysis
RBF   Radial basis function
TPWL Trajectory piecewise linearizaton
DD   Domain decomposition
FOM   Full-order model

1 Introduction

History matching is the process of calibrating uncertain reservoir model parameters such as gridblock permeabilities, porosities, faults multipliers and facies distributions,

✉ Cong Xiao
  C.Xiao@tudelft.nl

Extended author information available on the last page of the article.

through minimization of a cost function that quantifies the misfit between simulated and observed data (typically well data such as oil or water rates or bottom-hole pressure, but possibly also 4D seismic data). If the gradient of the cost function with respect to parameters can be computed using the adjoint of the reservoir model, history matching problems can be efficiently solved using a gradient-based minimization algorithm [11]. In general, significant effort is required to obtain and maintain a correct implementation of the adjoint model for complex nonlinear simulation models. Such implementations are generally intrusive, that is, they require access to the model code, which may not always be possible.

Many efforts have been taken to make the implementation of the adjoint model more feasible. One way is to replace the original complex model with a surrogate that the construction of the adjoint model becomes easier. Courtier et al. [12] proposed an incremental approach by replacing a high-resolution nonlinear model with an approximated linear model. Liu et al. [27, 28] developed an ensemble-based four-dimensional variational (4DEnVar) data assimilation scheme where the approximated linear model is

constructed using an ensemble of model forecasts. Recently, to extend the ensemble-based tangent linear model (TLM) to more realistic applications, Frolov and Bishop et al. [5, 14] incorporated a local ensemble tangent linear model (LETLM) into 4D-Var scheme. The LETLM has the ability to capture localized physical features of dynamic models with relatively small ensemble size. However, it will become intractable for high-dimensional systems. Proper Orthogonal Decomposition (POD), a model order reduction method, is a possible approach to decrease the dimensionality of the original model. The POD approach has been applied to various disciplines, including reservoir model simulations [21, 30], and has in some cases shown significant speed up [7] .

The combination of model linearization and model reduction techniques has the potential to further ease the implementation of adjoint models for high-dimensional complex dynamic systems. Vermeulen and Heemink [40] combined a non-intrusive perturbation-based linearization method and POD to build a reduced-order linear approximation of the original high-dimensional non-linear model. The adjoint of this reduced-order linear model can be easily constructed and therefore the minimization of the objective function can be handled efficiently. Altaf et al. [1] and Kaleta et al. [24] applied this method to a coastal engineering and reservoir history matching problem, respectively.

Alternatively, the Trajectory Piecewise Linearization (TPWL) can be classified as a model-intrusive linearization method. In TPWL, a number of full-order "training" runs is first simulated, and then a linear model is generated through first-order expansion around the "closest" training trajectories. In reservoir engineering, Cardoso et al. [8] were the first to integrate POD and TPWL methods and applied this strategy to oil production optimization. He et al. applied the POD-TPWL method to both reservoir history matching and production optimization [18, 20]. These studies suggested that POD-TPWL has the potential to significantly reduce the runtime for subsurface flow problems [19, 37]. A drawback, however, is that the POD-TPWL method requires access to derivative matrices used internally by the numerical solver and therefore cannot be used with most commercial simulators [20, 39]. And while non-intrusive reduced-order linear model construction is possible [1, 24, 40], the required derivative information is estimated using a global perturbation-based finite difference method, which needs a large number of full-order simulations and is therefore computationally demanding. Furthermore, the global perturbation also hinders the extension of this method to large-scale reservoir history matching which requires retaining many POD patterns [24]. In order to avoid model intrusion and numerous full-order simulations, we propose to incorporate domain decomposition (DD) and radial basis function (RBF) interpolation into POD-TPWL

to develop a new non-intrusive subdomain POD-TPWL algorithm.

RBF interpolation is mainly used to construct surrogate models [23] and has been applied for example to reservoir engineering and fluid dynamics [25, 42, 43]. Recently, Bruyelle et al. [6] applied the neural network-based RBF to obtain the first-order and second-order derivative information of a reservoir model and estimate the gradients and Hessian matrix for reservoir production optimization. The accuracy of RBF-based gradient approximation is determined by the sampling strategies of the interpolation data [6]. For high-dimensional problems, the classical global RBF interpolation algorithm requires a large number of interpolation data to capture the flow dynamic as much as possible [10]. Moreover, the global RBF algorithm can cause some spurious long-distance correlations, which implies the possibilities to avoid some redundant interpolation data. This motivates us to develop a subdomain RBF interpolation technique for reservoir models where the domain decomposition (DD) technique potentially allows us to apply the methodology to large-scale problems. Different local RBF interpolation schemes are considered based on the details of local flow dynamics in each subdomain. The domain decomposition technique first introduced in the work of Przemieniecki [35] has been applied to various fields [4]. Lucia et al. [29] first introduced the DD method to model-order reduction for accurately tracking a moving strong shock wave. Subsequently, the DD method has also been applied to non-linear model reduction problems [2, 3, 9].

This paper presents a new non-intrusive subdomain POD-TPWL algorithm for subsurface flow. The key idea behind this subdomain POD-TPWL is to integrate the DD method and RBF interpolation into a model linearization procedure based on POD-TPWL. The LETLM also enables us construct a tangent linear model with relatively small ensemble size. The key point of LETLM is to identify the influence area which is very similar to the purpose of domain decomposition described here. However, this LETLM needs to sequentially construct the TLM piecewise for each state variable, in this study, we construct the reduced-order linear model (TLM) piecewise for each subdomain instead of each state variable. After constructing the reduced-order linear model using the subdomain POD-TPWL algorithm, because of the linearity in the reduced-order subspace, the implementation of adjoint model is easy, and thus, it is convenient to incorporate this reduced-order linear model into a gradient-based reservoir history matching procedure. The runtime speedup and the accuracy of the new history matching algorithm have been assessed.

This paper is organized as follows. The history matching problem and the classical adjoint-based solution approach

are described in Section 2. Section 3 contains the mathematical background of the traditional POD-TPWL. Section 4 gives the mathematical descriptions of domain decomposition (DD) and radial basis function (RBF) interpolation, which are used to develop the non-intrusive subdomain POD-TPWL algorithm. In addition, a workflow for combining subdomain POD-TPWL with an adjoint-based history matching algorithm is described. Section 5 discusses and evaluates an application of the new history matching workflow to two numerical "twin" experiments involving synthetic reservoir models. Finally, Section 6 summarizes our contribution and discusses future work.

2 Problem description

A single simulation step of a discretized two-phase oil-water reservoir system is described as follows,

$$\mathbf{x}^{n+1} = \mathbf{f}^{n+1}(\mathbf{x}^n, \boldsymbol{\beta}, \boldsymbol{u}^{n+1}), \quad n = 0, \cdots, N-1 \tag{1}$$

where, $\mathbf{f}^{n+1}: R^{2N_g} \to R^{2N_g}$ represents the nonlinear time-dependent model evolution, $\mathbf{x}^{n+1} \in R^{2N_g}$ represents the state vector (pressure and saturation in every gridblock), N_g is the total number of gridblocks, n and $n+1$ indicate the timesteps, N denotes the total number of simulation steps, and $\boldsymbol{\beta}$ denotes the vector of uncertain parameters, which is the spatial permeability field in our case. $\boldsymbol{u}^{n+1}$ denotes the vector of well controls, e.g., bottom-hole pressure and injection or production rate. For the history matching problem, these well controls have been predefined. To simplify the notation, we will intentionally omit the well control terms. For more details about the discretization of the governing equations, see e.g., [34].

The relationship between simulated data $\mathbf{y}^{m+1}$, state vector $\mathbf{x}^{m+1}$, uncertain parameters $\boldsymbol{\beta}$ and well controls $\boldsymbol{u}^{n+1}$ is described by a nonlinear operator $\mathbf{h}^{m+1}: R^{2N_g} \to R^{N_d}$, which, in our case, represents the well model (for seismic data another model would be needed). N_d is the number of measurements at each timestep. The simulated measurements are therefore described by

$$\mathbf{y}^{m+1} = \mathbf{h}^{m+1}(\mathbf{x}^{m+1}, \boldsymbol{\beta}, \boldsymbol{u}^{m+1}), \quad m = 0, \cdots, N_0 - 1 \tag{2}$$

where N_0 is the number of timesteps at which measurements are taken.

The history matching process calibrates the uncertain parameters by minimizing a cost function defined as a sum of weighted squared differences between observed and modeled measurements (data). Additional incorporation of prior information into the cost function as a regularization term can further constrain the minimization procedure and make

the history matching problem well-posed [33]. Eventually, the cost function is described by the sum of two terms.

$$J(\mathbf{x}^1, \cdots, \mathbf{x}^n, \cdots, \mathbf{x}^N, \boldsymbol{\beta}) = \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_p)^T \mathbf{R}_p^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_p)$$
$$+ \frac{1}{2} \sum_{m=1}^{N_0} (\mathbf{d}_o^m - \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\beta}))^T \mathbf{R}_m^{-1}(\mathbf{d}_o^m - \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\beta})) \tag{3}$$

where $\mathbf{d}_o^m$ represents the vector of observed data at timestep m.

In validation experiments, $\mathbf{d}_o^m$ is generated by adding some noise, e.g., $\mathbf{r}^m$, to the data $\mathbf{y}_t^m$ simulated with a "truth" model. We will assume here that $\mathbf{r}^m$ is a time-dependent vector of observation errors at time level m, which is uncorrelated over time, and satisfies the Gaussian distribution $G(\mathbf{0}, \mathbf{R}_m)$ where $\mathbf{R}_m$ is the observation error covariance matrix at the timestep m. $\boldsymbol{\beta}_p$ represents the prior parameter vector, and $\mathbf{R}_p$ represents the error covariance matrix of the prior parameters, which characterizes the uncertainty in the prior model. A gradient-based optimization algorithm can be used to determine a parameter set that is not too far away from the prior information, while minimizing the misfit between the observed and simulated data.

The key step of a gradient-based minimization algorithm is to determine the gradient of the cost function with respect to the parameters. In the adjoint approach, a modified function $\hat{J}$ is obtained by adjoining the model equation Eq. 1 to $\mathbf{J}$

$$\hat{J}(\mathbf{x}^1, \cdots, \mathbf{x}^n, \cdots, \mathbf{x}^N, \boldsymbol{\beta}) = J(\mathbf{x}^1, \cdots, \mathbf{x}^n, \cdots, \mathbf{x}^N, \boldsymbol{\beta})$$
$$+ \sum_{n=1}^{N} [\boldsymbol{\lambda}^n]^T [\mathbf{x}^n - \mathbf{f}^n(\mathbf{x}^{n-1}, \boldsymbol{\beta})] \tag{4}$$

And then, the gradient of the cost function is formulated after introducing the adjoint model as follows (more details about the mathematical derivation can be found in [22]),

$$[\frac{dJ}{d\boldsymbol{\beta}}]^T = \mathbf{R}_p^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_p) - \sum_{n=1}^{N} [\boldsymbol{\lambda}^n]^T \frac{\partial \mathbf{f}^n}{\partial \boldsymbol{\beta}}$$
$$- \sum_{m=1}^{N_0} \left[\frac{\partial \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right]^T \mathbf{R}_m^{-1}(\mathbf{d}_o^m - \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\beta})) \tag{5}$$

where the adjoint model in terms of the Lagrange multipliers $\boldsymbol{\lambda}^n$ is given by

$$\boldsymbol{\lambda}^n = \left[\frac{\partial \mathbf{f}^{n+1}}{\partial \mathbf{x}^n} \right] \boldsymbol{\lambda}^{n+1} + \left[\frac{\partial \mathbf{h}^n(\mathbf{x}^n, \boldsymbol{\beta})}{\partial \mathbf{x}^n} \right]^T \mathbf{R}_n^{-1}(\mathbf{d}_o^n - \mathbf{h}^n(\mathbf{x}^n, \boldsymbol{\beta})) \tag{6}$$

for $n = N, \cdots 1$ with an ending condition $\boldsymbol{\lambda}^{N+1} = 0$. This adjoint approach has a high computational efficiency because just one forward simulation and one backward simulation are required to compute the gradient, independent on the size of the variable vector. It should

be pointed out that four derivative terms, e.g., $\frac{\partial \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$, $\frac{\partial \mathbf{h}^n(\mathbf{x}^n, \boldsymbol{\beta})}{\partial \mathbf{x}^n}$, $\frac{\partial \mathbf{f}^n}{\partial \boldsymbol{\beta}}$ and $\frac{\partial \mathbf{f}^{n+1}}{\partial \mathbf{x}^n}$, are required in the adjoint method. We will give detailed descriptions of how to efficiently obtain these four terms using our proposed subdomain POD-TPWL algorithm in the following sections.

3 POD-TPWL algorithm

In the TPWL scheme, one or more full-order "training" runs using a set of perturbed parameters are simulated first. The states and the derivative information at each time step from these runs are used to construct the TPWL surrogate. Given the state $\mathbf{x}^n$ and parameters $\boldsymbol{\beta}$, the state $\mathbf{x}^{n+1}$ is approximated as a first-order expansion around the training solution $(\mathbf{x}_{tr}^{n+1}, \mathbf{x}_{tr}^n, \boldsymbol{\beta}_{tr})$ as follows,

$$\mathbf{x}^{n+1} \approx \mathbf{x}_{tr}^{n+1} + \mathbf{E}^{n+1}(\mathbf{x}^n - \mathbf{x}_{tr}^n) + \mathbf{G}^{n+1}(\boldsymbol{\beta} - \boldsymbol{\beta}_{tr}) \quad (7)$$

where

$$\mathbf{E}^{n+1} = \frac{\partial \mathbf{f}^{n+1}}{\partial \mathbf{x}_{tr}^n}, \quad \mathbf{G}^{n+1} = \frac{\partial \mathbf{f}^{n+1}}{\partial \boldsymbol{\beta}_{tr}} \quad (8)$$

The training solution $(\mathbf{x}_{tr}^{n+1}, \mathbf{x}_{tr}^n, \boldsymbol{\beta}_{tr})$ is chosen to be as "close" as possible to the state $\mathbf{x}^n$. A detailed description of a criterion for closeness can be found in [17]. The matrices $\mathbf{E}^{n+1} \in R^{2N_g \times 2N_g}$ and $\mathbf{G}^{n+1} \in R^{2N_g \times N_g}$ represent the derivative of the dynamic model as Eq. 1 at timestep $n+1$ with respect to states $\mathbf{x}_{tr}^n$ and parameters $\boldsymbol{\beta}_{tr}$ respectively. Equation 7 is, however, still in a high-dimensional space, e.g., $\mathbf{x}^{n+1} \in R^{2N_g}$, and $\boldsymbol{\beta} \in R^{N_g}$, which motivates the development of the POD-TPWL algorithm [17].

POD provides as means to project the high-dimensional states into an optimal lower-dimensional subspace. The basis of this subspace is obtained by performing a Singular Value Decomposition (SVD) of a snapshot matrix containing the solution states at selected time steps (snapshots) computed from training simulations. The state vector $\mathbf{x}$ can then be represented in terms of the product of a coefficient vector $\boldsymbol{\psi}$ and a matrix of basis vectors $\boldsymbol{\Phi}$

$$\mathbf{x} = \boldsymbol{\Phi}\boldsymbol{\psi} \quad (9)$$

Let $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Phi}_s$ represent separate matrices of basis vectors for pressure and saturation respectively. In general, there is no need to contain all columns of the left singular matrix in $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Phi}_s$ and a reduced state vector representations are obtained by selecting only the first columns according to an energy criterion; see, e.g., [17]. To normalize the reduced state vector, the columns of $\boldsymbol{\Phi}_p$ are determined by multiplying the left singular matrix $\mathbf{U}_p$ with the singular value matrix $\boldsymbol{\Sigma}_p$ (and similarly for saturation), i.e.,

$$\boldsymbol{\Phi}_p = \mathbf{U}_p \boldsymbol{\Sigma}_p, \quad \boldsymbol{\Phi}_s = \mathbf{U}_s \boldsymbol{\Sigma}_s \quad (10)$$

These two matrix are then assembled into a single basis matrix $\boldsymbol{\Phi}$ as follows:

$$\mathbf{x} = \boldsymbol{\Phi}\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{\Phi}_p & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_s \end{bmatrix} \boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{\Phi}_p & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_s \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}_p \\ \boldsymbol{\psi}_s \end{bmatrix} \quad (11)$$

In this paper, we also use Karhunen-Loeve expansion (KLE) or principal component analysis (PCA) to parameterize the parameter space. KLE reduces the dimension of the parameter vector by projecting the high-dimensional parameter into an optimal lower-dimensional subspace [15]. The basis of this subspace is obtained by performing an eigenvalue decomposition of the prior parameter covariance matrix $\mathbf{R}_p$. If this covariance matrix is not accessible, the basis can alternatively be obtained from an SVD of a matrix holding an ensemble of prior parameter realizations with ensemble mean $\boldsymbol{\beta}_b = \boldsymbol{\beta}_p$. Including normalization of the reduced parameter vector, a random parameter vector sample $\boldsymbol{\beta}$ is generated as follows,

$$\boldsymbol{\beta} = \boldsymbol{\beta}_b + \boldsymbol{\Phi}_\beta \boldsymbol{\xi}, \quad \text{with} \quad \boldsymbol{\Phi}_\beta = \mathbf{U}_\beta \boldsymbol{\Sigma}_\beta \quad (12)$$

where $\boldsymbol{\Phi}_\beta$ denotes the matrix of parameter basis vectors, $\mathbf{U}_\beta$ and $\boldsymbol{\Sigma}_\beta$ are the matrix of left singular vectors and singular value matrix respectively, and $\boldsymbol{\xi}$ denotes a vector with independent Gaussian random variables with zeros mean and unit variance. A reduced parameter space representation is obtained by selecting only the first several columns of $\boldsymbol{\Phi}_\beta$. The number of retained columns for basis matrix (denoted as l_p and l_s for pressure and saturation, l_β for parameter, respectively) is determined through an energy criterion [17]. We take $\boldsymbol{\Phi}_p$ as an example. We first compute the total energy E_t, which is defined as $E_t = \sum_{i=1}^L v_i^2$, where v_i denotes the i-th diagonal element of singular value matrix $\boldsymbol{\Sigma}_p$. The energy associated with the first l_p singular vectors is given by $E_{l_p} = \sum_{i=1}^{l_p} v_i^2$. Then, the smallest l_p is determined such that E_{l_p} exceeds a specific fraction of E_t, e.g., 95%. The same procedure is applied to determine l_s and l_β.

Substituting Eqs. 11 and 12 into Eq. 7, we obtain the following POD-TPWL formula,

$$\boldsymbol{\psi}^{n+1} \approx \boldsymbol{\psi}_{tr}^{n+1} + \mathbf{E}_{\boldsymbol{\psi}}^{n+1}(\boldsymbol{\psi}^n - \boldsymbol{\psi}_{tr}^n) + \mathbf{G}_{\boldsymbol{\xi}}^{n+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \quad (13)$$

$$\mathbf{E}_{\boldsymbol{\psi}}^{n+1} = \boldsymbol{\Phi}^T \frac{\partial \mathbf{f}^{n+1}}{\partial \mathbf{x}_{tr}^n} \boldsymbol{\Phi}, \quad \mathbf{G}_{\boldsymbol{\xi}}^{n+1} = \boldsymbol{\Phi}^T \frac{\partial \mathbf{f}^{n+1}}{\partial \boldsymbol{\beta}_{tr}} \boldsymbol{\Phi}_\beta \quad (14)$$

Similarly, the well model Eq. 2 is also linearized around a close training solution $(\boldsymbol{\psi}_{tr}^{n+1}, \boldsymbol{\xi}_{tr})$ in the reduced space as follows,

$$\mathbf{y}^{m+1} \approx \mathbf{y}_{tr}^{m+1} + \mathbf{A}_{\boldsymbol{\psi}}^{m+1}(\boldsymbol{\psi}^{m+1} - \boldsymbol{\psi}_{tr}^{m+1}) + \mathbf{B}_{\boldsymbol{\xi}}^{m+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \quad (15)$$

$$\mathbf{A}_{\boldsymbol{\psi}}^{m+1} = \frac{\partial \mathbf{h}^{m+1}}{\partial \mathbf{x}_{tr}^{m+1}} \boldsymbol{\Phi}, \quad \mathbf{B}_{\boldsymbol{\xi}}^{m+1} = \frac{\partial \mathbf{h}^{m+1}}{\partial \boldsymbol{\beta}_{tr}} \boldsymbol{\Phi}_{\boldsymbol{\beta}} \qquad (16)$$

Equations 13 and 15 represent the POD-TPWL system for reservoir model and well model in the reduced-order space, respectively. In general, the traditional POD-TPWL method modifies the source code to output all derivative matrices [17]. In this paper, radial basis function interpolation is used to approximately estimate these derivative matrices. These derivative matrices then are substituted into POD-TPWL algorithm to form a subdomain reduced-order linear model.

4 Adjoint-based history matching using subdomain reduced-order linear model

This section describes the mathematical background of domain decomposition (DD), and radial basis function (RBF) interpolation, which are used to construct a non-intrusive subdomain reduced-order linear model. And then a procedure to incorporate this reduced-order linear model into an adjoint-based history matching is given in the last subsection.

4.1 Domain decomposition method

We denote a 2D or 3D computational domain as Ω. The entire domain Ω is assumed to be decomposed into S non-overlapping subdomains Ω^d, $d \in \{1, 2, \cdots, S\}$ (such that $\Omega = \bigcup_{d=1}^{S} \Omega^d$ and $\Omega^i \cap \Omega^j = 0$ for $i \neq j$). Each subdomain has local unknowns, e.g., local pressure and saturation variables. In each subdomain Ω^d, the generated snapshots within that subdomain are used to construct a set of local POD basis functions $\boldsymbol{\phi}^d$ and the corresponding POD coefficients $\boldsymbol{\psi}^{d,n+1}$ at the timestep $n+1$ as described in the previous section. The dynamic model Eq. 1 is replaced by an interpolation model relating neighboring subdomain POD coefficients at the current and previous time step.

$$\boldsymbol{\psi}^{d,n+1} = \boldsymbol{\pounds}^{d,n+1}(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) \qquad (17)$$

Similarly, the dynamic well model Eq. 2 is replaced by a second interpolation model expressed in terms of the local subdomain POD coefficient $\boldsymbol{\psi}^{d,m+1}$ and $\boldsymbol{\xi}$

$$\mathbf{y}^{d,m+1} = \hbar^{d,n+1}\left(\boldsymbol{\psi}^{d,m+1}, \boldsymbol{\xi}\right) \qquad (18)$$

where, vector $\boldsymbol{\psi}^{d,n}$ denotes the set of POD coefficients at the time level n for the subdomain Ω^d, $\boldsymbol{\psi}^{sd,n+1}$ denotes the set of POD coefficients at time level $n+1$ for the surrounding subdomains Ω^{sd}. In a 2-D case, the number of surrounding subdomains associated to subdomain Ω^d is between 2 and 4. Figure 1 shows a maximum of four surrounding subdomains connected with the subdomain Ω^5, three surrounding subdomains connected with the subdomain Ω^2, Ω^4, Ω^6, Ω^8, and two surrounding subdomains connected with the subdomain Ω^1, Ω^3, Ω^7, Ω^9.

We propose to use RBF interpolation to obtain the derivative matrices that are required by the POD-TPWL. In addition, domain decomposition has the abilities to efficiently capture localized physical features [10] and therefore has the potential to improve the derivative estimation by local low-dimensional RBF interpolation which will be described in the next subsections.

4.2 Radial basis function interpolation

RBF interpolation can be classified as a data-driven interpolation method [23, 25, 42]. High-dimensional interpolation needs a large number of data to obtain a satisfactory accuracy, a phenomenon often referred to as the "curse of dimensionality". To remedy this difficulty, DD approximates the global domain by the sum of the local subdomains, and therefore can be applied to form a locally low-dimensional RBF interpolation.

For subdomain Ω^d, let $\boldsymbol{\pounds}^{d,n+1}(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi})$ denote a RBF interpolation function for the POD coefficient $\boldsymbol{\psi}^{d,n+1}$ at the time level $n+1$. The RBF interpolation function is a linear combination of M radial basis functions in the form of,

$$\boldsymbol{\pounds}^{d,n+1}(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) =$$
$$and \quad \sum_{j=1}^{M} \boldsymbol{\omega}_j^{d,n+1} \times \theta(||(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi})$$
$$-(\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)||) \qquad (19)$$

where, $\boldsymbol{\omega}^{d,n+1}$ is a weighting coefficient vector of size M. $||(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)||$ is a scalar distance using L^2 norm. θ is a set of specific radial basis functions.

The specific coefficient $\boldsymbol{\omega}_j^{d,n+1}$ is determined so as to ensure that the interpolation function values $\boldsymbol{\pounds}^{d,n+1}$ at the training data points $(\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)$, match the given data $\boldsymbol{\psi}_j^{d,n+1}$ exactly. This can be expressed by,

$$\mathbf{D}^{d,n+1}\boldsymbol{\omega}^{d,n+1} = \mathbf{Z}^{d,n+1} \qquad (20)$$

where

$$\mathbf{D}^{d,n+1} = \begin{bmatrix} \theta(l^{n+1}(1,1)) & ... & \theta(l^{n+1}(1,M)) \\ . & \theta(l^{n+1}(i,j)) & . \\ \theta(l^{n+1}(M,1)) & ... & \theta(l^{n+1}(M,M)) \end{bmatrix}$$
$$l^{n+1}(i,j) = ||(\boldsymbol{\psi}_i^{d,n}, \boldsymbol{\psi}_i^{sd,n+1}, \boldsymbol{\xi}_i) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)||,$$
$$i = 1, \cdots, M; \ j = 1, \cdots, M \qquad (21)$$

Fig. 1 Illustration of domain decomposition in a 2-D case
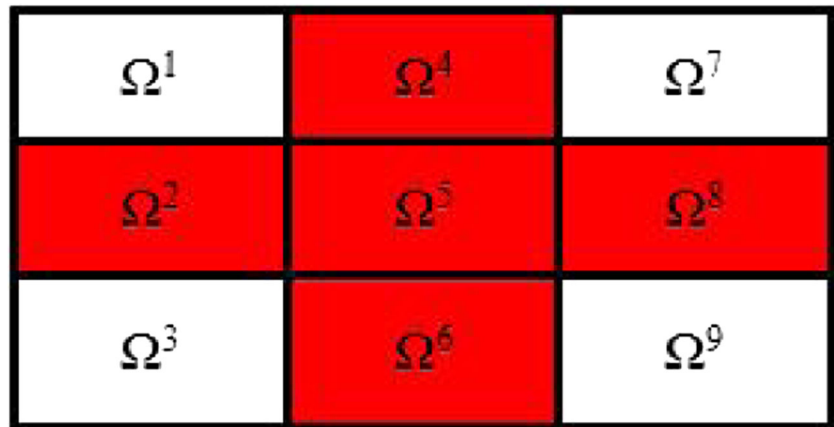
$$\boldsymbol{\omega}^{d,n+1} = [\boldsymbol{\omega}_1^{d,n+1}, \boldsymbol{\omega}_2^{d,n+1}, \cdots, \boldsymbol{\omega}_M^{d,n+1}]^T \quad (22)$$

$$\mathbf{Z}^{d,n+1} = [\boldsymbol{\psi}_1^{d,n+1}, \boldsymbol{\psi}_2^{d,n+1}, \cdots, \boldsymbol{\psi}_M^{d,n+1}]^T \quad (23)$$

The weighting coefficients are determined by solving the linear system Eq. 20. We chose Multi-Quadratic RBF in our case. l represents a Euclidean distance. ϵ denotes the shape parameters, which can be optimized using greedy algorithm [43]. Clearly, a Multi-Quadratic RBF monotonically decreases with Euclidean distance. Multi-Quadratic RBF is local and is the msot commonly used across several applications. Other type of RBF θ can be chosen with more specific purpose [41]. A list of well-known RBF are provided in Table 1.

After the construction of RBF interpolation, we can analytically estimate the derivative at the "closest" training data point, e.g., $(\boldsymbol{\psi}_i^{d,n}, \boldsymbol{\psi}_i^{sd,n+1}, \boldsymbol{\xi}_i)$, by differentiating the RBF as follows,

$$\frac{\partial \boldsymbol{\mathcal{L}}^{d,n+1}}{\partial \boldsymbol{\xi}}\Big|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} = \sum_{j=1}^M \boldsymbol{\omega}_j^{d,n+1} \times$$
$$\frac{\partial \theta(\|(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)\|)}{\partial \boldsymbol{\xi}}\Big|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} \quad (24)$$

$$\frac{\partial \boldsymbol{\mathcal{L}}^{d,n+1}}{\partial \boldsymbol{\psi}^{sd,n+1}}\Big|_{\boldsymbol{\psi}^{sd,n+1}=\boldsymbol{\psi}_i^{sd,n+1}} = \sum_{j=1}^M \boldsymbol{\omega}_j^{d,n+1} \times$$
$$\frac{\partial \theta(\|(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)\|)}{\partial \boldsymbol{\psi}^{sd,n+1}}\Big|_{\boldsymbol{\psi}^{sd,n+1}=\boldsymbol{\psi}_i^{sd,n+1}} \quad (25)$$

$$\frac{\partial \boldsymbol{\mathcal{L}}^{d,n+1}}{\partial \boldsymbol{\psi}^{d,n}}\Big|_{\boldsymbol{\psi}^{d,n}=\boldsymbol{\psi}_i^{d,n}} = \sum_{j=1}^M \boldsymbol{\omega}_j^{d,n+1} \times$$
$$\frac{\partial \theta(\|(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)\|)}{\partial \boldsymbol{\psi}^{d,n}}\Big|_{\boldsymbol{\psi}^{d,n}=\boldsymbol{\psi}_i^{d,n}} \quad (26)$$

Similarly, the approximation Eq. 18 also can be constructed using RBF interpolation method as follows,

$$\mathbf{y}^{d,m+1} \approx \hbar^{d,n+1}(\boldsymbol{\psi}^{d,m+1}, \boldsymbol{\xi})$$
$$= \sum_{j=1}^M \boldsymbol{\varepsilon}_j^{d,m+1} \times \theta(\|(\boldsymbol{\psi}^{d,m+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,m+1}, \boldsymbol{\xi}_j)\|) \quad (27)$$

The derivative at the training data by differentiating the RBF function Eq. 17 with respect to $(\boldsymbol{\psi}_i^{d,m+1}, \boldsymbol{\xi}_i)$ can be given by

$$\frac{\partial \hbar^{d,n+1}}{\partial \boldsymbol{\xi}}\Big|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} = \sum_{j=1}^M \boldsymbol{\varepsilon}_j^{d,m+1} \times$$
$$\frac{\partial \theta(\|(\boldsymbol{\psi}^{d,m+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,m+1}, \boldsymbol{\xi}_j)\|)}{\partial \boldsymbol{\xi}}\Big|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} \quad (28)$$

$$\frac{\partial \hbar^{d,n+1}}{\partial \boldsymbol{\psi}^{d,m+1}}\Big|_{\boldsymbol{\psi}^{d,m+1}=\boldsymbol{\psi}_i^{d,m+1}} = \sum_{j=1}^M \boldsymbol{\varepsilon}_j^{d,m+1} \times$$
$$\frac{\partial \theta(\|(\boldsymbol{\psi}^{d,m+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,m+1}, \boldsymbol{\xi}_j)\|)}{\partial \boldsymbol{\xi}}\Big|_{\boldsymbol{\psi}^{d,m+1}=\boldsymbol{\psi}_i^{d,m+1}} \quad (29)$$

where $\boldsymbol{\varepsilon}^{d,m+1}$ is a weighting coefficient vector of size M (number of training data sets), and these matrices are reconstructed for each time step correspondingly.

Table 1 Some well-known radial basis functions

Functions	Definition
Gaussian	$\theta(l) = e^{-(\frac{l}{\epsilon})^2}$
Linear Spline	$\theta(l) = l$
Multi-Quadratic	$\theta(l) = \sqrt{l^2 + \epsilon^2}$
Inverse Caddric	$\theta(l) = \frac{1}{l^2 + \epsilon^2}$
Cubic Spline	$\theta(l) = l^3$
Thin Plate Spline	$\theta(l) = l^2 log l$
Inverse Multistory	$\theta(l) = \frac{1}{\sqrt{l^2 + \epsilon^2}}$

4.3 Subdomain POD-TPWL algorithm

By considering the dynamic interaction between neighboring subdomains as in Eq. 17, the coefficients $\boldsymbol{\psi}^{d,n+1}$ for the subdomain Ω^d can be obtained as follows,

$$
\begin{aligned}
\boldsymbol{\psi}^{d,n+1} \approx {} & \boldsymbol{\psi}_{tr}^{d,n+1} + \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n+1}(\boldsymbol{\psi}^{d,n} - \boldsymbol{\psi}_{tr}^{d,n}) \\
& + \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n+1}(\boldsymbol{\psi}^{sd,n+1} - \boldsymbol{\psi}_{tr}^{sd,n+1}) + \mathbf{G}_{\boldsymbol{\xi}_{tr}}^{n+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \quad (30)
\end{aligned}
$$

Coupling domain decomposition and radial basis function interpolation, the derivative matrices required by POD-TPWL for the subdomain Ω^d are estimated as follows

$$
\begin{aligned}
\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n+1} &\approx \frac{\partial \boldsymbol{\pounds}^{d,n+1}}{\partial \boldsymbol{\psi}^{d,n}}\big|_{\boldsymbol{\psi}^{d,n}=\boldsymbol{\psi}_{tr}^{d,n}} \\
\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n+1} &\approx \frac{\partial \boldsymbol{\pounds}^{d,n+1}}{\partial \boldsymbol{\psi}^{sd,n+1}}\big|_{\boldsymbol{\psi}^{sd,n+1}=\boldsymbol{\psi}_{tr}^{sd,n+1}} \\
\mathbf{G}_{\boldsymbol{\xi}}^{n+1} &\approx \frac{\partial \boldsymbol{\pounds}^{d,n+1}}{\partial \boldsymbol{\xi}}\big|_{\boldsymbol{\xi}=\boldsymbol{\xi}_{tr}} \quad (31)
\end{aligned}
$$

where subindex tr refers to the nearest training point determined in terms of of vector $(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi})$.

Similarly, substituting Eq. 27-Eq. 28 into Eq. 15, the simulated measurements $\mathbf{y}^{d,m+1}$ of the subdomain Ω^d are reformulated as

$$
\mathbf{y}^{d,m+1} \approx \mathbf{y}_{tr}^{d,m+1} + \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m+1}(\boldsymbol{\psi}^{d,m+1} - \boldsymbol{\psi}_{tr}^{d,m+1}) + \mathbf{B}_{\boldsymbol{\xi}_{tr}}^{m+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \quad (32)
$$

$$
\mathbf{A}_{\boldsymbol{\psi}}^{d,m+1} \approx \frac{\partial \hbar^{d,m+1}}{\partial \boldsymbol{\psi}^{d,m+1}}\big|_{\boldsymbol{\psi}^{d,m+1}=\boldsymbol{\psi}_{tr}^{d,m+1}}, \ B_{\boldsymbol{\xi}_{tr}}^{m+1} \approx \frac{\partial \hbar^{d,m+1}}{\partial \boldsymbol{\xi}}\big|_{\boldsymbol{\xi}=\boldsymbol{\xi}_{tr}} \quad (33)
$$

The implementation of POD-TPWL is local in each subdomain, which has the potential to capture features dominated by local dynamics better than global approximations. Therefore, we could refer to the subdomain POD-TPWL algorithm. The subdomain POD-TPWL consists of an offline stage and an online stage. (1) During the offline stage, we construct a set of local RBF and estimate the derivative information for each subdomain. Firstly, the solutions of the full-order model are saved as a sequence of snapshots over the whole computational domain and then partitioned into subdomains. From the local sequence of

snapshots over each subdomain, a number of local basis vectors is formed using POD. Unlike the traditional practices in which RBF is used to construct a set of surrogates for each subdomain, we use RBF to estimate the derivative matrices for each subdomain. Finally, those estimated derivative matrices are substituted into POD-TPWL algorithm to form a reduced-order linear model in each subdomain. (2) The online stage consists of the time evolution of the dynamic state of the reduced model by iterative implementation and solution of the subdomain POD-TPWL equations. Referring to Eq. 30, we represent the dynamic interactions among neighboring subdomains using an implicit formula. the variables of one subdomain at current time level can be linearized around the variables of this subdomain at previous timestep and variables of neighboring subdomains at current timestep, which have not been determined. Thus, additional iterations are required. The non-adjacent subdomains almost have no direct dynamic interactions, this kind of subdomain POD-TPWL algorithm can be easily parallelized. Referring to Fig. 1, subdomains $\Omega^1, \Omega^3, \Omega^5, \Omega^7, \Omega^9$ have no direct interactions, and therefore the subdomain POD-TPWL algorithm can be simultaneously implemented in these five subdomains. This is similar for the subdomains $\Omega^2, \Omega^4, \Omega^6, \Omega^8$. This parallelization for subdomain POD-TPWL is known as block red-black ordering [13, 16]. The k-th iterative description of Eq. 29 is as follows.

$$
\begin{aligned}
\boldsymbol{\psi}_k^{d,n+1} \approx {} & \boldsymbol{\psi}_{tr}^{d,n+1} + \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n+1}(\boldsymbol{\psi}^{d,n} - \boldsymbol{\psi}_{tr}^{d,n}) \\
& + \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n+1}(\boldsymbol{\psi}_{k-1}^{sd,n+1} - \boldsymbol{\psi}_{tr}^{sd,n+1}) + \mathbf{G}_{\boldsymbol{\xi}_{tr}}^{n+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \quad (34)
\end{aligned}
$$

The iteration for subdomain POD-TPWL is very cheap, thus, we do not limit the maximum number of iterations. The iteration will be stopped when no further changes in the estimate of state, .e.g., pressure and saturation occur,

$$
\frac{|\boldsymbol{\psi}^{k+1} - \boldsymbol{\psi}^k|}{max\{|\boldsymbol{\psi}^{k+1}|, 1\}} < \eta_{\boldsymbol{\psi}} \quad (35)
$$

Nonetheless, the parallelization is not explored in this paper and is left as a future area of research.

Fig. 2 The illustration of the reconstruction method of subdomain POD-TPWL algorithm

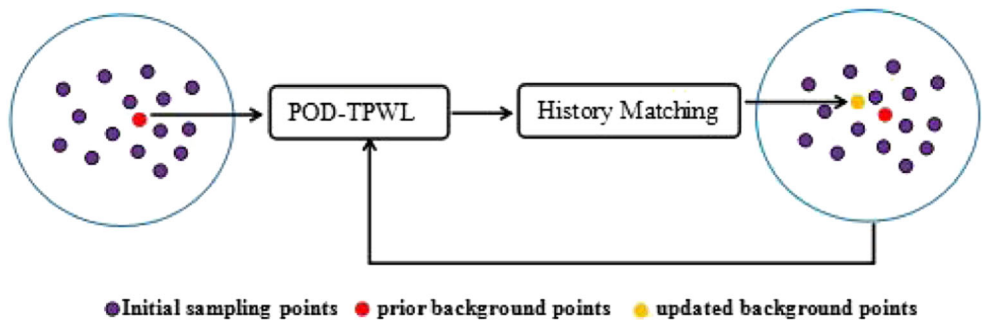● Initial sampling points ● prior background points ● updated background points

Fig. 3 The illustration of adjoint-based history matching using subdomain POD-TPWL algorithm. The purple box represents the inner loop in which the model-reduced cost function as Eq. 36 is minimized. The left arrow represents the outer-loop in which the reduced-order linear model is updated
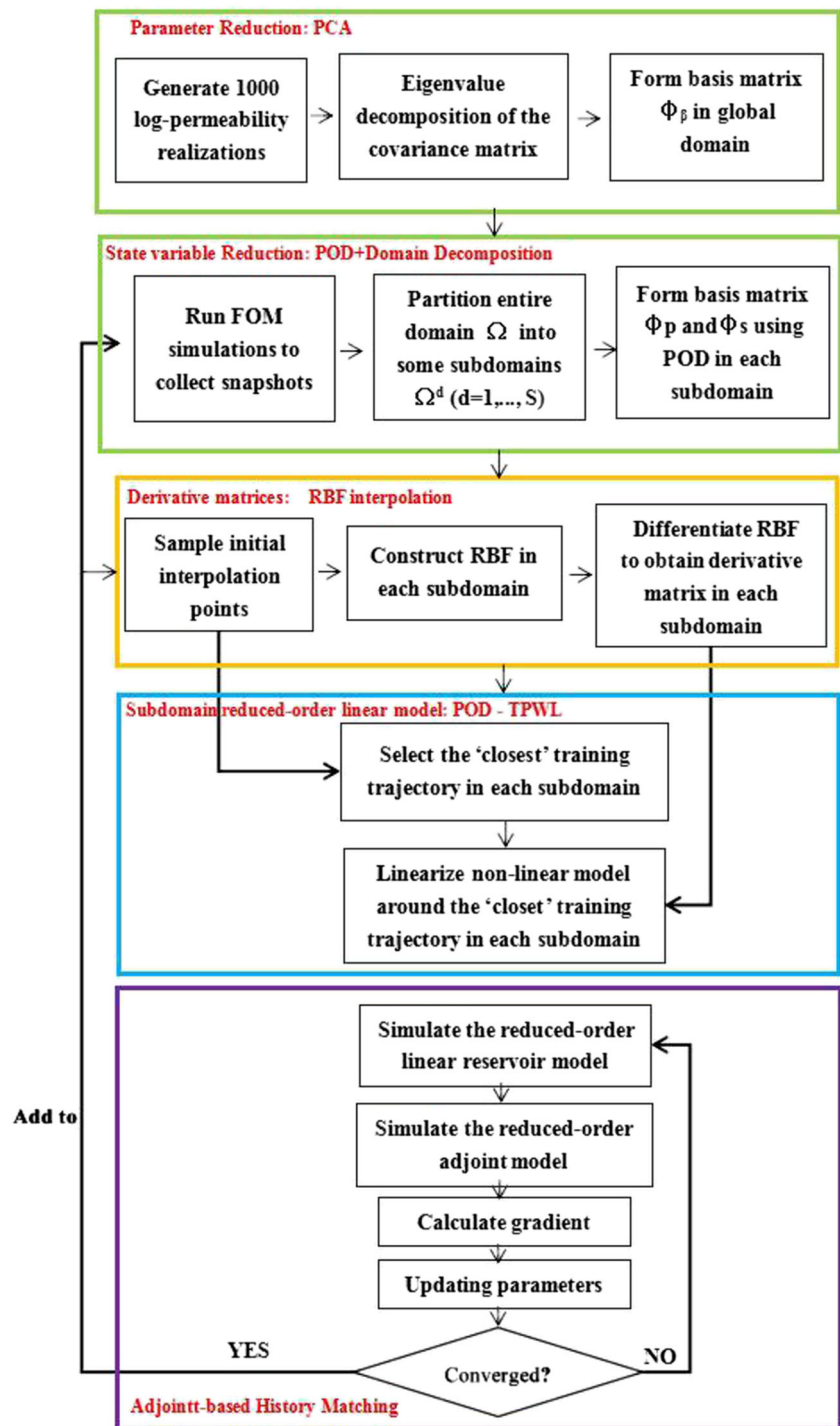
4.4 Sampling strategy

In our proposed subdomain POD-TPWL algorithm, training points are required for both RBF interpolation and to construct the POD basis. For POD, the snapshot matrix generated from the training simulations should accurately characterize the dynamic behavior of the system. The training simulations used to construct the RBF interpolation model should allow for accurate computation of derivative matrices. The procedure for choosing these training points will be described here.
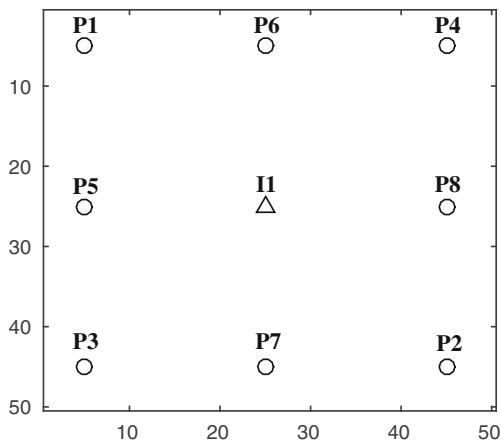
Fig. 4 The well placement in the 2-D reservoir model for case 1

Sampling strategy for POD. A small initial set of model parameter vectors is sampled and used as input for full-order model (FOM) simulations from which a snapshot matrix is constructed. The singular value spectrum is computed for this initial set of samples. The number of samples is then increased one at a time, i.e., adding one FOM simulation, and the SVD is recomputed, until no significant changes are observed in the singular value spectrum.

Sampling strategy for RBF. The accuracy of the RBF interpolation will be reduced if too few data points are chosen, while the computational cost increases with the

Table 2 Experiment settings using MRST for case 1

Description	Value
Dimensions	$50 \times 50 \times 1$
Grid cell size	$20 \times 20 \times 10$
Number of wells	8 producers, 1 injector
Fluid density	1014 kg/m^3, 859 kg/m^3
Fluid viscosity	0.4 mP·s, 2 mP·s
Initial pressure	30 MPa
Initial saturation	S_o=0.80, S_w=0.20
Connate water saturation	S_{wc}=0.20
Residual oil saturation	S_{or}=0.20
Corey exponent, oil	4.0
Corey exponent, water	4.0
Injection rate	200m^3/d
BHP	25MPa
History production time	5 year
Prediction time	10 year
Timestep	0.1 year
Measurement timestep	0.2 year
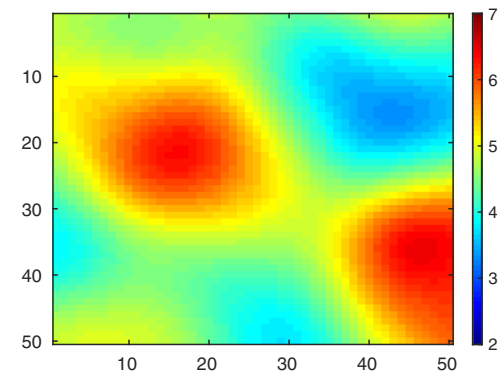
(a) 'True' model in original full-order space
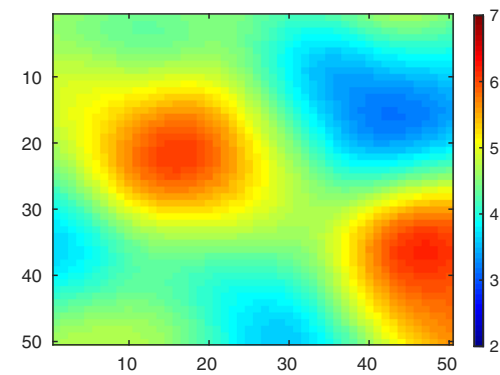
(b) Projected 'true' model in reduced-order subspace

Fig. 5 Comparison of the "true" reservoir model in full-order space and in reduced-order space for Case 1

number of data points, which will be prohibitive if too many points are chosen. To limit the number of FOM simulations used to construct the interpolation model for the POD coefficients, we use 2-sided perturbation of each coefficient ξ_j resulting in $2 \times l_\beta + 1$ points. In some experiments, we add additional points by simultaneous random sampling of perturbations $\Delta \boldsymbol{\xi}$. An alternative could be use to use Smolyak sparse grid sampling [36].

4.5 Adjoint-based history matching algorithm

After constructing the linear reduced-order model using the proposed subdomain POD-TPWL methodology, it can be used within an adjoint-based history matching workflow.

The cost function evaluated using the reduced-order linear model is given by reformulating Eq. 3 as follows,

$$
\begin{aligned}
J(\boldsymbol{\xi}) = {} & \frac{1}{2}(\boldsymbol{\beta}_b + \boldsymbol{\Phi}_\beta \boldsymbol{\xi} - \boldsymbol{\beta}_p)^T \mathbf{R}_p{}^{-1}(\boldsymbol{\beta}_b + \boldsymbol{\Phi}_\beta \boldsymbol{\xi} - \boldsymbol{\beta}_p) \\
& + \frac{1}{2} \sum_{d=1}^{S} \sum_{m=1}^{N_0} [\mathbf{d}_o^{d,m} - \mathbf{y}_{tr}^{d,m} - \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m}(\boldsymbol{\psi}^{d,m} - \boldsymbol{\psi}_{tr}^{d,m}) - \mathbf{B}_{\boldsymbol{\xi}_{tr}}^m(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})]^T \\
& \mathbf{R}_m{}^{-1}[\mathbf{d}_o^{d,m} - \mathbf{y}_{tr}^{d,m} - \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m}(\boldsymbol{\psi}^{d,m} - \boldsymbol{\psi}_{tr}^{d,m}) - \mathbf{B}_{\boldsymbol{\xi}_{tr}}^m(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})] \quad (36)
\end{aligned}
$$

Fig. 6 Examples of realizations of the log-permeability field generated from PCA coefficients sampled randomly from the set {-1, 1} for Case 1
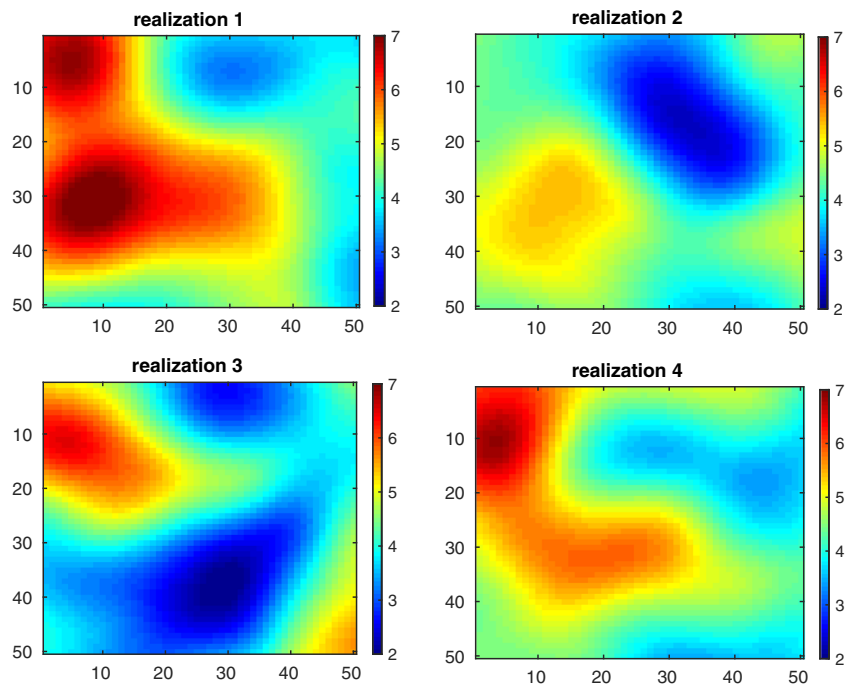
After augmenting this cost function with the model equation Eq. 29, the gradient with respect to the model parameters is obtained as

$$[\frac{dJ}{d\boldsymbol{\xi}}]^T = (\boldsymbol{\phi_\beta})^T \mathbf{R}_p^{-1}(\boldsymbol{\beta}_b + \boldsymbol{\phi_\beta}\boldsymbol{\xi} - \boldsymbol{\beta}_p)$$

$$- \sum_{d=1}^{S}\sum_{m=1}^{N_0}[\mathbf{B}_{\boldsymbol{\xi}_{tr}}^m]^T \mathbf{R}_m^{-1}[\mathbf{d}_o^{d,m} - \mathbf{y}_{tr}^{d,m} - \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m}(\boldsymbol{\psi}^{d,m} - \boldsymbol{\psi}_{tr}^{d,m})$$

$$- \mathbf{B}_{\boldsymbol{\xi}_{tr}}^m(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})] - \sum_{d=1}^{S}\sum_{n=1}^{N}[\mathbf{G}_{\boldsymbol{\xi}_{tr}}^n]^T \boldsymbol{\lambda}^{d,n} \qquad (37)$$

where $\boldsymbol{\lambda}^{d,n}$ is obtained as the solution of the adjoint model for the subdomain Ω^d as follows

$$[\mathbf{I} - (\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n})^T]\boldsymbol{\lambda}^{d,n} = \sum_{d=1}^{S}[\mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,n}]^T \mathbf{R}_n^{-1}[\mathbf{d}_o^{d,n} - \mathbf{y}_{tr}^{d,n}$$

$$- \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,n}(\boldsymbol{\psi}^{d,n} - \boldsymbol{\psi}_{tr}^{d,n}) - \mathbf{B}_{\boldsymbol{\xi}_{tr}}^n(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})] + [\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n}]^T \boldsymbol{\lambda}^{d,n+1} \qquad (38)$$

The minimization of the cost function Eq. 36 is performed using a steepest descent algorithm [32] and is stopped when either one of the following stopping criteria is satisfied

– No further changes in the cost function

$$\frac{|J(\boldsymbol{\xi}^{k+1}) - J(\boldsymbol{\xi}^k)|}{max\{|J(\boldsymbol{\xi}^{k+1})|, 1\}} < \eta_J \qquad (39)$$

– No further changes in the estimate of parameters,

$$\frac{|\boldsymbol{\xi}^{k+1} - \boldsymbol{\xi}^k|}{max\{|\boldsymbol{\xi}^{k+1}|, 1\}} < \eta_{\boldsymbol{\xi}} \qquad (40)$$

– The maximum number of iterations has been reached. i.e

$$k <= N_{max} \qquad (41)$$

where η_J and $\eta_{\boldsymbol{\xi}}$ are predefined error constraints and N_{max} is the maximum number of iterations.

As mentioned in [24], the solution of the reduced and linearized minimization problem based on Eq. 36 is not necessarily the solution of the original problem based on Eq. 3. Therefore, an additional stopping criterion should be introduced for the original model as follows [38],

$$N_d N_0 - 2\sqrt{2N_d N_0} \leqslant 2J(\boldsymbol{\beta}^k) \leqslant N_d N_0 + 2\sqrt{2N_d N_0} \quad (42)$$

Table 3 Experiment design of error quantification for case 1

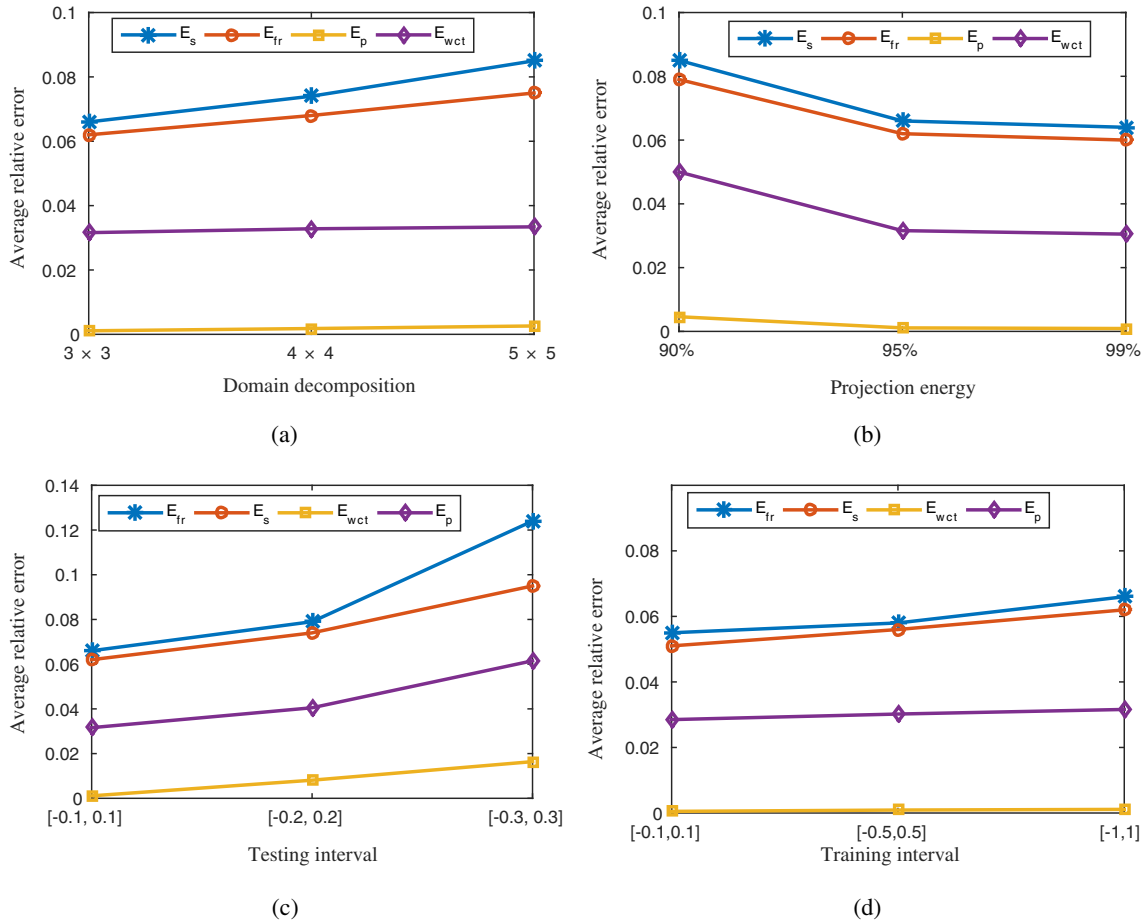| Domain decomposition	Projection energy	Testing interval	training interval
3×3	90%	[−0.1,0.1]	[−0.1,0.1]
4×4	95%	[−0.2,0.2]	[−0.5,0.5]
5×5	99%	[−0.3,0.3]	[−1,1]

(a)

(b)

(c)

(d)

Fig. 7 Average subdomain POD-TPWL errors for test case as a function of domain decomposition, projection energy, testing interval and training interval for case 1. Results are for TPWL models constructed using 37 training simulations

where, N_0 is the number of timesteps where the measurements are taken, N_d is the number of measurements at each timestep, $\boldsymbol{\beta}^k$ represents the updated parameters vector at the k-th outer-loop. J is the cost function computed as Eq. 3.

If the objective function does not meet the stopping criterion as Eq. 42, then additional outer-loops are required

to reconstruct new reduced-order linear models using the updated parameters, and the aforementioned iterative inner-loop is performed again.

Table 4 Summary of the number of reduced variables for the global domain and after domain decomposition for case 1 (Note: **s** refers to saturation, **p** refers to pressure)

Fig. 8 Illustration of the applied domain decomposition for Case 1

Domain Decomposition				Global Domain		
SD	β	s	p	β	s	p
1		14	7			
2		13	6			
3		12	5			
4		13	4			
5	18	16	7	18	72	36
6		14	6			
7		13	5			
8		15	6			
9		12	5			
Total	18	122	51	18	72	36
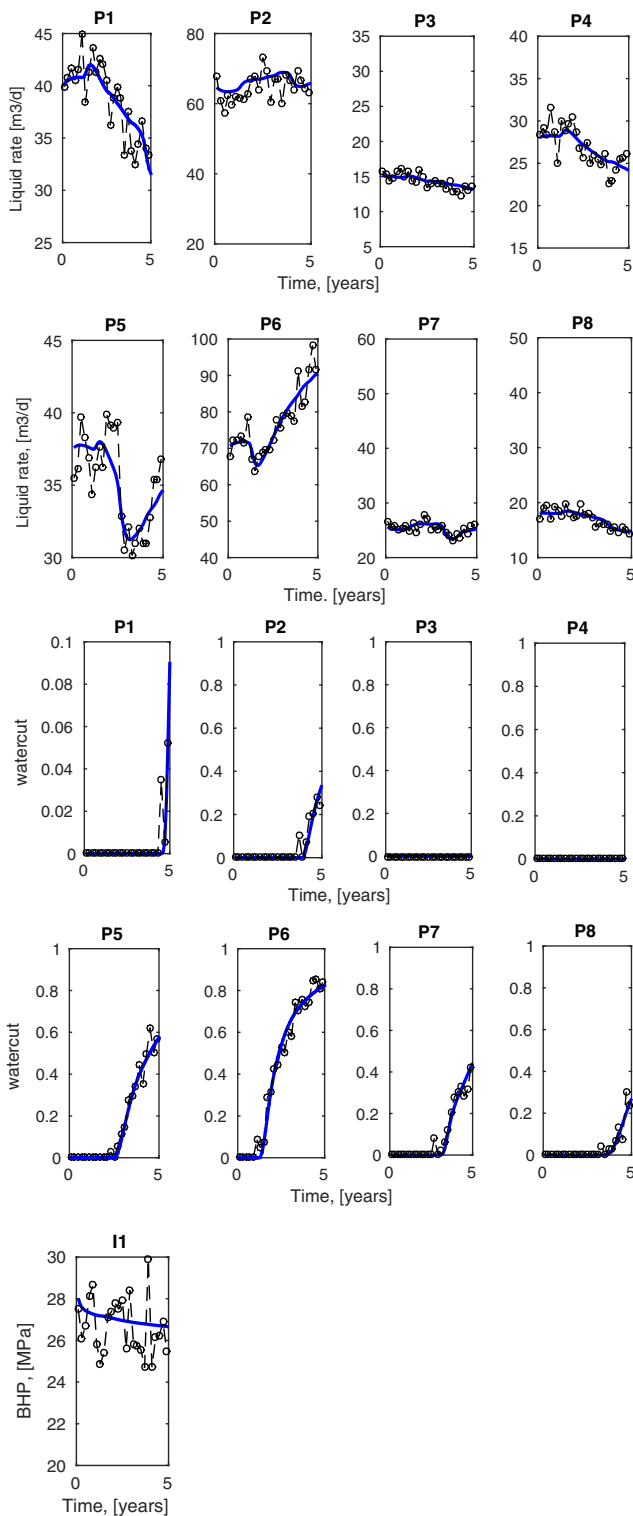
Fig. 9 Measured quantities for case 1. Blue solid line: reference model (truth), black dashed line: noisy data

Our proposed non-intrusive subdomain POD-TPWL has computational advantages over the traditional construction of reduced-order linear models using perturbation-based finite-difference method proposed in [24], especially

when the reduced-order linear model is required to be reconstructed for each outer-loop. Instead of re-perturbing the parameter and state variables one by one to approximate the derivative matrices as proposed in [24], which would require $(l_p + l_s + l_\beta + 1)$ FOM simulations, our algorithm runs only one additional FOM simulation using updated parameters. The updated parameters and simulated snapshots are added into the previous group of sampling interpolation points and corresponding snapshots. The derivative matrices for the updated parameters are approximated based on the updated group of interpolation points and snapshots. The overall workflow has been summarized conceptually in Fig. 2. The individual steps of the history matching algorithm described in this section are summarized in the flow chart presented in Fig. 3.

5 Numerical experiments and discussion

In this section, two numerical experiments are presented that aim to demonstrate and evaluate our proposed adjoint-based history matching algorithm. The first experiment is based on a small 2D synthetic model containing 9 wells. The second experiment uses a reservoir model with 13 wells based on the SAIGUP benchmark case [31]. In our numerical experiments, MRST, a free open-source software for reservoir modeling and simulation [26], is used to run the FOM simulations.

5.1 Case 1 - 2D reservoir with 9 wells

5.1.1 Description of model settings

A 2D heterogeneous oil-water reservoir is considered with two-phase imcompressible flow dynamics. The reservoir contains 8 producers and 1 injector, which are labeled as P_1 to P_8, and I_1 respectively, see Fig. 4. Detailed information about the reservoir geometry, rock properties, fluid properties, and well controls is summarized in Table 2.

5.1.2 Reduced model construction

We generate an ensemble of 1000 Gaussian-distributed realizations of log-permeability. We also assume that the generated log-permeability fields are not conditioned to the permeability values at the well locations. The log-permeability fields and the corresponding porosity fields are described by the following statistics:

$$\sigma_\beta = 5 \tag{43}$$

$$\mathbf{C}_\beta(x_{i1,j1}; y_{i2,j2}] = \sigma_\beta^2 e^{-\left[\left(\frac{|x_{i1}-x_{i2}|}{\chi_x}\right)^2 + \left(\frac{|y_{i1}-y_{i2}|}{\chi_y}\right)^2\right]} \tag{44}$$

Fig. 10 True, initial and updated log-permeability fields using SD POD-TPWL, GD POD-TPWL, and the FD method for case 1
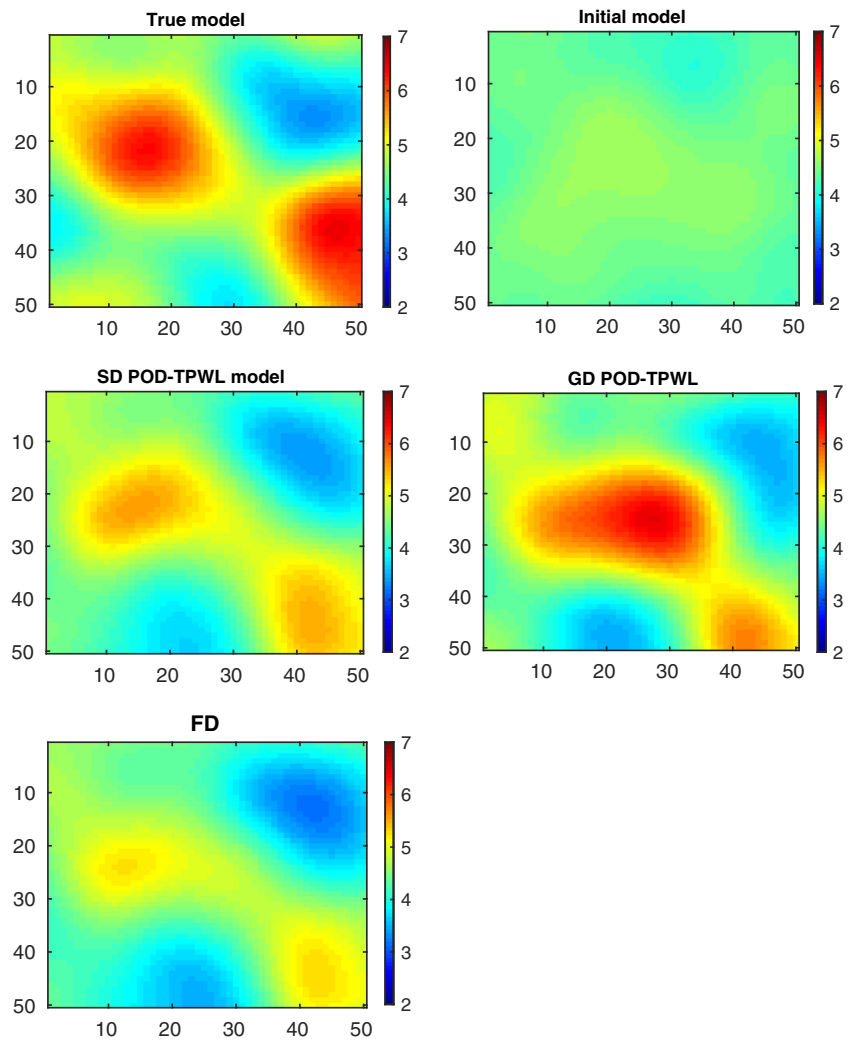
$$\frac{\chi_x}{L_x} = 0.2, \frac{\chi_y}{L_y} = 0.2 \tag{45}$$

$$\boldsymbol{\phi} = 0.25 \left(\frac{e^{\beta}}{200} \right)^{0.1} \tag{46}$$

Here, σ_{β} is the standard deviation of log-permeability $\boldsymbol{\beta}$; C_{β} is the covariance of $\boldsymbol{\beta}$; $x_{i1,j1}=(x_{i1}, y_{j1})$ denotes the coordinates of a grid block; χ_x (or χ_y) is the correlation length in x (or y) direction; and L_x (or L_y) is the domain length in x (or y) direction. The background log-permeability $\boldsymbol{\beta}_b$ is taken as the average of the 1000 realizations. One of the realizations was considered to be the truth, and is illustrated in Fig. 5a. The permeability field was parameterized using KL-expansion and about 95% energy is maintained, resulting in 18 permeability patterns with $l_{\beta} = 18$ corresponding independent PCA coefficients, which are used in the workflow as a low-dimensional representation of the 2500 grid block permeability values.

Figure 5b shows the projection of the "true" permeability field in this low-dimensional subspace which shows that the truth can be almost perfectly reconstructed in this subspace. Four realizations for log-permeability field generated are additionally shown in Fig. 6.

After having reduced the parameter space, the next step is to reduce the reservoir model. The first step is to generate a set of training runs from which snapshots will be taken. Since the required number of training runs is not known a priori we follow the following procedure: (1) generate a sample PCA coefficient vector by sampling from the set $\{-1, 1\}$, (2) run a full-order model simulation with these parameters, (3) extract snapshots and form the snapshot matrix, (4) compute the singular value decomposition of the snapshot matrix (5) repeat steps (1) to (4) until changes in the singular values are insignificant. For Case 1, this produced a set of 15 training runs and 240 snapshots for pressure and saturation each.
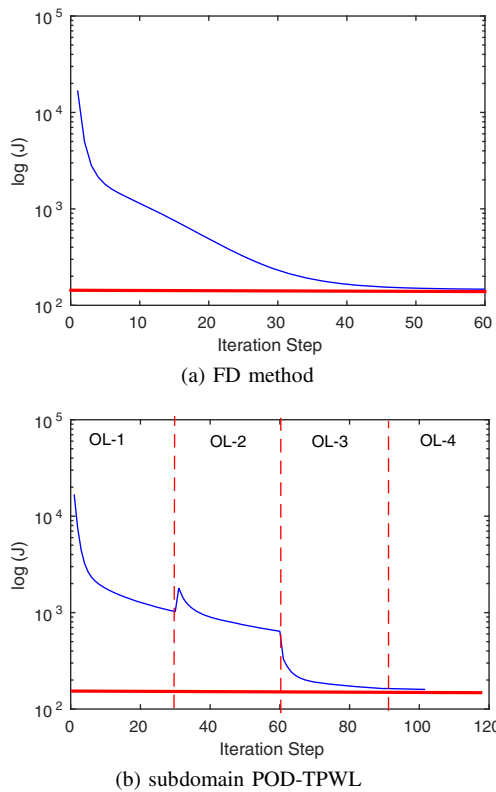
(a) FD method

(b) subdomain POD-TPWL

Fig. 11 Cost function value decrease using **a** finite-difference method, and **b** subdomain POD-TPWL for case 1. OL-i means the i-th outer-loop

5.1.3 Error quantifications

The performance of the subdomain POD-TPWL model can be investigated by comparing the errors relative to FOM simulation for quantities of interest. Here, errors are quantified in terms of the mismatch of the fluid rate, water-cut and primal variables, i.e., pressure and saturation between the FOM solution d_{FOM} and subdomain POD-TPWL simulations d_{ROM}. For example, the average fluid rate error E_{fr} or the average water-cut error E_{wct} is calculated as

$$E_{fr} \quad \text{or} \quad E_{wct} = \frac{1}{NN_d} \sum_{i=1}^{N} \sum_{j=1}^{N_d} \frac{|(\mathbf{d}_{FOM}^{i,j} - \mathbf{d}_{ROM}^{i,j})|}{\mathbf{d}_{FOM}^{i,j}} \quad (47)$$

where $\mathbf{d}$ represents the fluid rate or water-cut. Similarly, the pressure error E_p and saturation error E_s are formulated as

$$E_p \quad \text{or} \quad E_s = \frac{1}{NN_d} \sum_{i=1}^{N} \sum_{j=1}^{N_d} \frac{|(\mathbf{x}_{FOM}^{i,j} - \mathbf{x}_{ROM}^{i,j})|}{\mathbf{x}_{FOM}^{i,j}} \quad (48)$$

where $\mathbf{x}$ represents the saturation or pressure in each gridblock at each timestep.
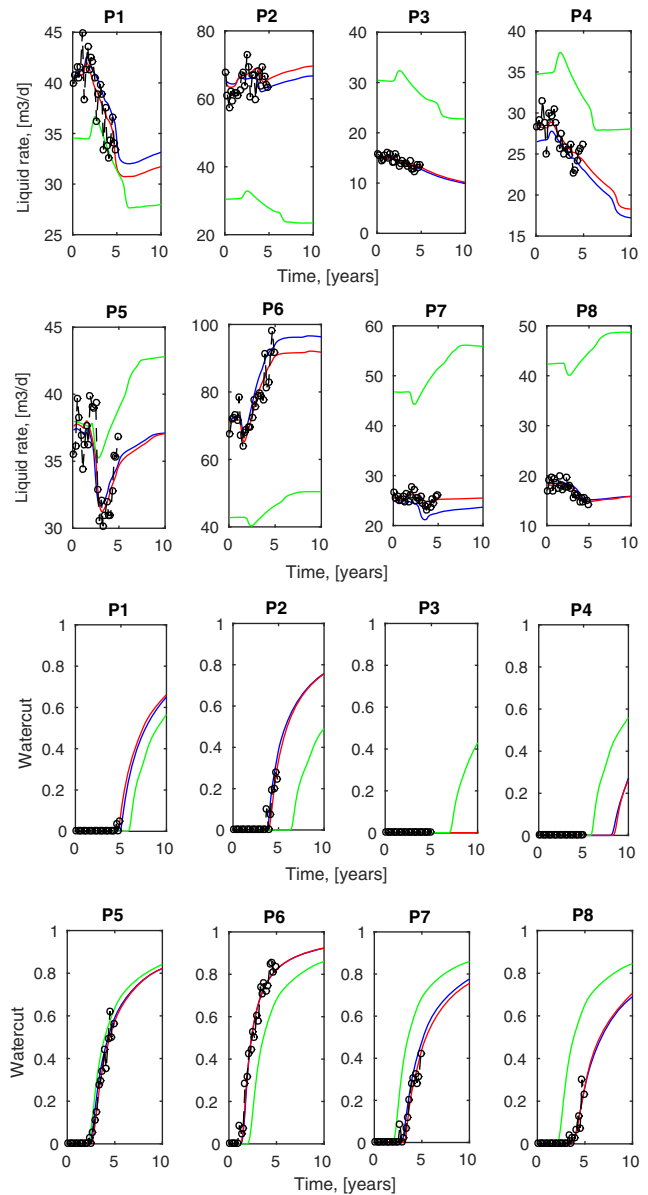
Fig. 12 Forecasts of the liquid rate and water-cut for case 1 from the initial model (green line), the reference model (blue line), and the model updated using the SD POD-TPWL (red line). Measured data are indicate by open circles

In our case, the initial 37 sampling points are selected as described in subsection of sampling strategy, where the j-th element ξ_i^j of the i-th PCA coefficient vector ξ^j is perturbed sequentially in 2 opposite directions (positive and negative) by a specific amplitude perturbation $\Delta \xi_i^j$. These 37 sampling points are used to build a subdomain reduced-order linear model. Four factors, e.g., domain decomposition, projection energy, testing interval and training interval, are considered to analyze the sensitivity of model errors. The experiment settings

Table 5 Comparison between SD POD-TPWL and FD method for case 1

–	Iterations	FOM	$J(\xi) \times 10^4$	e_{obs}	e_β
Initial model	–	–	1.69	28.38	2.28
SD POD-TPWL	103	55	0.0160	3.35	0.68
FD	52	988	0.0153	3.28	0.72
"True" model	–	–	0.0068	2.12	0

are shown in Table 3. We specify the stopping criterion $\eta_\psi = 10^{-3}$.

The detailed information about the error quantification can be found in our Supplementary file, the main results are summarized here. Figure 7 shows the error in fluid rate, water-cut, pressure and saturation as a function of the four factors. A relatively small subdomain size of 3 x 3 cells produced the most accurate results for this case. Accuracy is also improved by increasing the energy threshold and retaining more POD patterns, albeit at an increased computational cost. Retaining 95% of the total energy during projection produces an acceptable accuracy in this case. Increasing the testing interval, which represents the maximum discrepancy between test model and linearized training model, severely deteriorates the reduced model accuracy, with the best results obtained with the [–0.1, 0.1] interval. an appropriate iteration step-length for the history matching process should be set as 0.1 based on our numerical experiments.

In terms of computational effort, the runtime for a single FOM simulation for this case was about 2 s on a machine with i5-4690 Intel CPUs (4 cores, 3.5GHz) and 24 GB memory using Matlab-R2015a. The subdomain POD-TPWL models, in contrast, required less than 0.2 s. However, the construction of subdomain POD-TPWL requires simulating 52 training models, POD, derivative estimation using RBF, plus additional overhead, which severely increases the cost. Therefore, it would not make

sense to construct the subdomain POD-TPWL model unless it is to be used for a large number of simulations. Because many simulations are required in history matching applications, the subdomain POD-TPWL model should be applicable in this context. The use of subdomain POD-TPWL in conjunction with an adjoint-based data assimilation procedure is presented in the following parts.

### 5.1.4 History matching procedure

Based on the error sensitivity analysis presented above, we divide the entire domain into 9 (3 × 3) rectangle subdomains as illustrated in Fig. 8. The choice of subdomains is fairly arbitrary at this point since we have no formal algorithm to determine the best number and design of the subdomains. The previously collected global snapshots for pressures and saturations are divided into local snapshots. For each subdomain, two separate eigenvalue problems for pressure and saturation are solved using POD. The number of reduced parameters and state patterns for each subdomain and for the global domain are listed in Table 4 where specific projection energy, e.g., 95% and 95%, are preserved for the pressure and saturation respectively in each subdomain.

The history period is 5 years during which observations are taken from 8 producers and 1 injector every second simulation time step (nearly 73 days) resulting in 25 time instances. Noisy observations are generated from the model with the "true" permeability field and include bottom-hole pressures (BHP) in the injector and fluid rates and water-cut (WCT) in the producers. As a result, we have 200 fluid rates and 200 WCT values measured in the producers and 25 bottom-hole pressures measured in the injector, which gives in total 425 measurement data. Normal distributed independent measurement noise with a standard deviation equal to 5% of the "true" data value, was added to all observations. The generated measurements are shown in Fig. 9.

Table 6 The number of interpolation variables in each subdomain and global domain for case 1. Ω^d is the d-th subdomain

Domain decomposition		Global domain
1	$75 = 21(\Omega^1) + 19(\Omega^2) + 17(\Omega^4) + 18$	
2	$98 = 21(\Omega^1) + 19(\Omega^2) + 17(\Omega^3) + 23(\Omega^5) + 18$	
3	$74 = 19(\Omega^2) + 17(\Omega^3) + 20(\Omega^6) + 18$	
4	$97 = 21(\Omega^1) + 17(\Omega^4) + 23(\Omega^5) + 18(\Omega^7) + 18$	
5	$118 = 19(\Omega^2) + 17(\Omega^4) + 23(\Omega^5) + 20(\Omega^6) + 21(\Omega^8) + 18$	$126 = 72 + 36 + 18$
6	$95 = 17(\Omega^3) + 23(\Omega^5) + 20(\Omega^6) + 17(\Omega^9) + 18$	
7	$74 = 17(\Omega^4) + 18(\Omega^7) + 21(\Omega^8) + 18$	
8	$97 = 23(\Omega^5) + 18(\Omega^7) + 21(\Omega^8) + 17(\Omega^9) + 18$	
9	$76 = 20(\Omega^6) + 21(\Omega^8) + 17(\Omega^9) + 18$	

Fig. 13 Water saturation and
pressure fields from the
full-order model and from SD
POD-TPWL and GD
POD-TPWL based models for
case 1

(a) Full-order model

(b) Domain decomposition

(c) Global domain
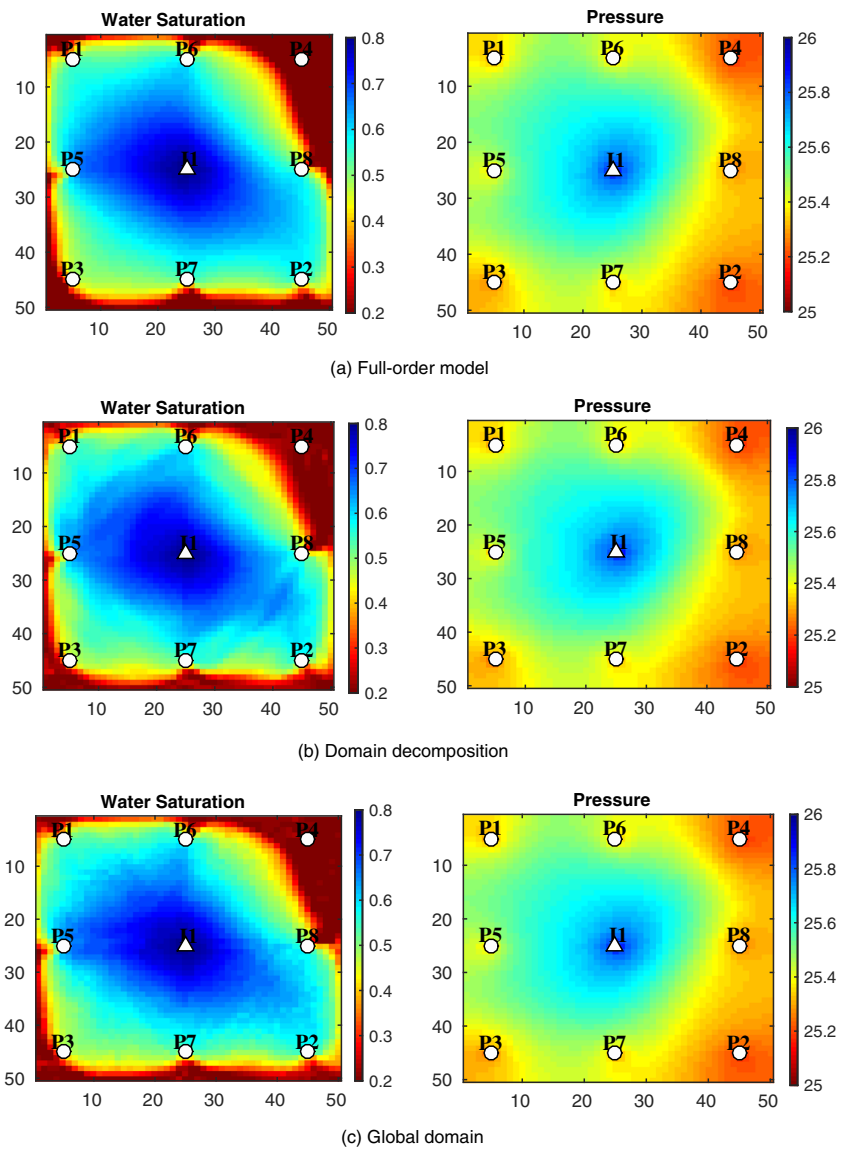
To analyze the results, we define two error measures
based on data misfits e_{obs} and parameter misfits e_{β} as
follows,

$$e_{obs} = \sqrt{\frac{\sum_{i=1}^{N_o} \sum_{j=1}^{N_d} (\mathbf{d}_{obs}^{i,j} - \mathbf{d}_{upt}^{i,j})^2}{N_o N_d}} \quad (49)$$

$$e_{\beta} = \sqrt{\frac{\sum_{i=1}^{N_g} (\boldsymbol{\beta}_{true}^i - \boldsymbol{\beta}_{upt}^i)^2}{N_g}} \quad (50)$$

where, $\mathbf{d}_{obs}^{i,j}$ and $\mathbf{d}_{upt}^{i,j}$ represent the measurements and
simulated data using the updated model respectively; $\boldsymbol{\beta}_{true}^i$
and $\boldsymbol{\beta}_{upt}^i$ denote the grid block log-permeability from the
"true" model and updated model respectively.

Figures 10, 11, and 12 and Table 5 show the results
of the first numerical experiments, including the updated
log-permeability field, the value of cost function at

each iteration and the mismatch between observed data
and predictions. To demonstrate the performance of our
proposed methodology, we compared the results with those
of finite-difference (FD)-based history matching algorithm
without domain decomposition and model order reduction.
The total computational cost of any minimization problem

Table 7 Comparison between SD POD-TPWL and GD POD-TPWL
for case 1

–	FOM	$J(\boldsymbol{\xi}) \times 10^4$	e_{obs}	e_{β}
Initial model	–	1.69	28.38	2.28
SD POD-TPWL	55	0.0160	3.35	0.68
GD POD-TPWL	73	0.0140	3.21	0.61
FD	988	0.0153	3.28	0.72
"True" model	–	0.0068	2.12	0

Fig. 14 Ensemble of 20 updated log-permeability fields for case 1
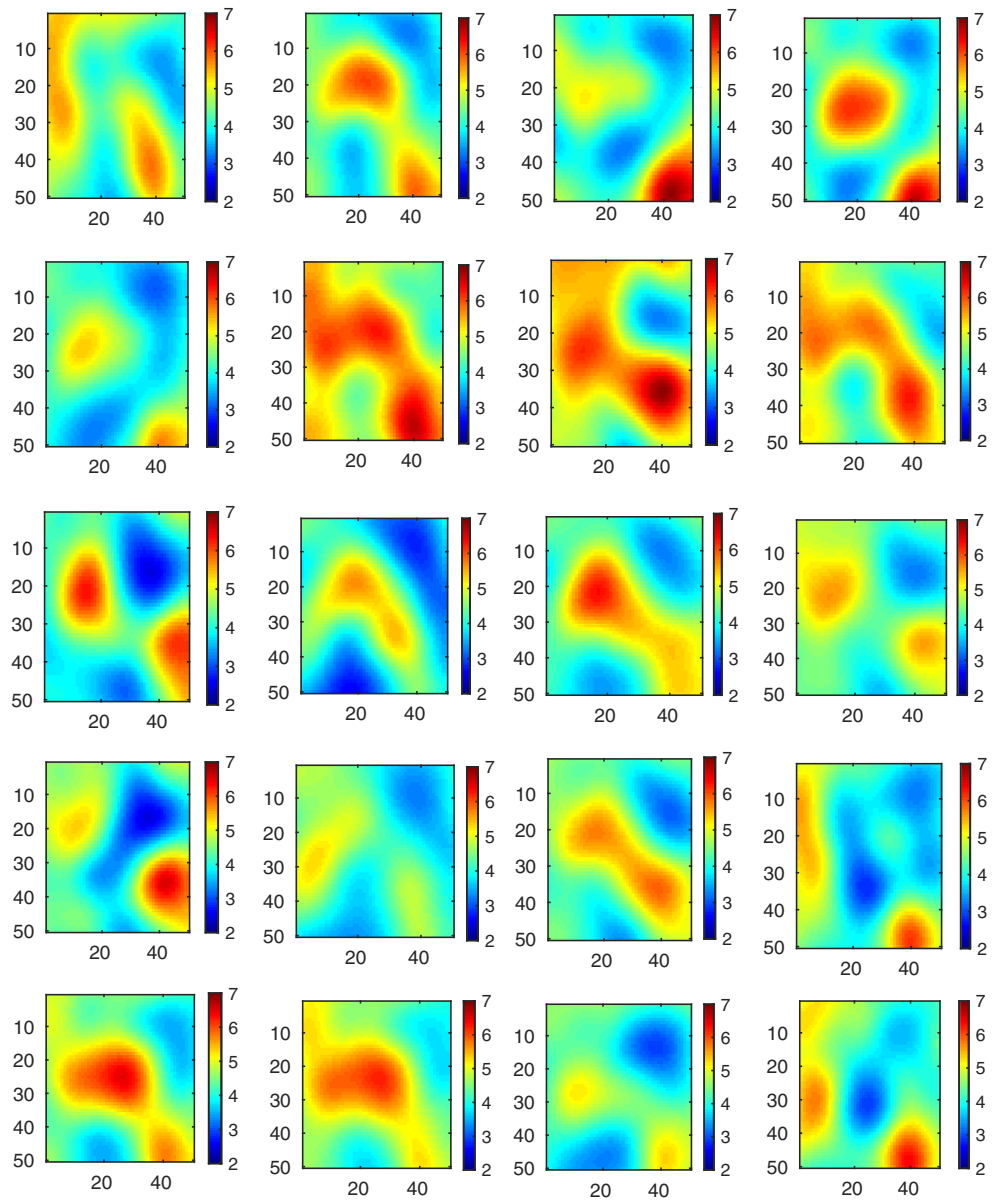
strongly depends on the number of parameters. In our work, for a fair comparison, we use the same parameterization to reduce the number of parameters and implement FD based history matching in this reduced-order parameter subspace. The cost function for FD based history matching can be defined as follows,

$$J(\boldsymbol{\xi}) = \frac{1}{2}(\boldsymbol{\beta}_b + \boldsymbol{\phi}_{\boldsymbol{\beta}}\boldsymbol{\xi} - \boldsymbol{\beta}_p)^T \mathbf{R}_p^{-1}(\boldsymbol{\beta}_b + \boldsymbol{\phi}_{\boldsymbol{\beta}}\boldsymbol{\xi} - \boldsymbol{\beta}_p)$$
$$+ \frac{1}{2}\sum_{m=1}^{N_0}(\mathbf{d}_o^m - \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\xi}))^T \mathbf{R}_m^{-1}(\mathbf{d}_o^m - \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\xi})) \quad (51)$$

The FD method is used to compute the numerical gradient of the cost function as Eq. 51 with respect to 18 PCA coefficients. A FD gradient is determined by one-sided perturbation of each of the 18 PCA coefficients. Thus, 19 full-order model (FOM) simulations are required for each iteration step. The stopping criteria are set $\eta_J = 10^{-4}$, $\eta_{\boldsymbol{\xi}} = 10^{-3}$, and N_{max}=30. As can be seen from Fig. 11 and Table 5, the model-reduced approach needs 55 FOM simulations, among them, 15 FOM simulations are used to collect the snapshots and 37 FOM simulations are used to construct the initial reduced-order linear model in the first outer-loop. The remaining 3 FOM simulations are used to reconstruct three new reduced-order linear models in the next three outer-loops and to calculate the cost function as Eq. 3 in the original space. Figure 10 shows the true, initial, and final estimates of log-permeability field. In this case, the main geological structures of the the "true" model can be reconstructed with both methods. However, the
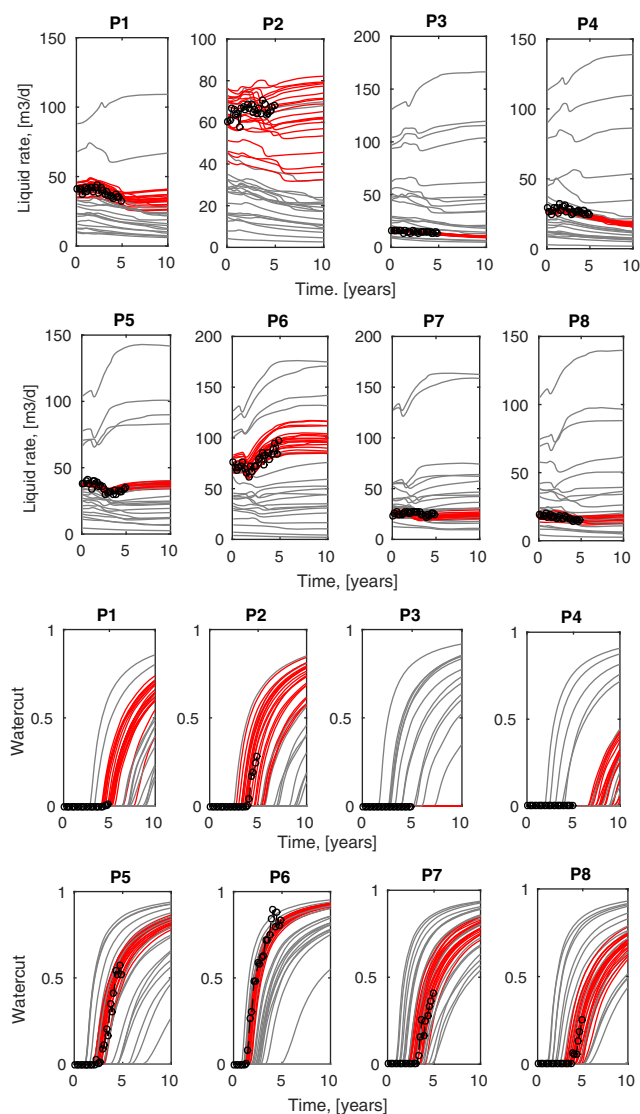
Fig. 15 Ensemble of fluid rate and water-cut prediction for case 1. The gray lines represent the predictions from the 20 prior permeability realizations, while the red lines represent the predictions from the corresponding 20 posterior permeability realizations calibrated using our method. The circles represent the noisy data

parameter estimates obtained with proposed methodology more accurately reproduce the true amplitudes than those obtained with FOM-based history matching. From Fig. 12 and Table 5, we can both qualitatively and quantitatively observe that the history matching process results in an improved prediction in all of the eight production wells. Figure 12 illustrates the data match of fluid rate and water-cut up to year 5 and an additional 5-year prediction until year 10 for all 8 producers. The prediction based on the initial model is far from that of the true model. After history matching, the predictions from the updated model match the observations very well. Also, the prediction of water breakthrough time is improved for all of the production

wells, also for wells that show water breakthrough only after the history period.

One of the key issues for the subdomain POD-TPWL is the implementation of the domain decomposition technique. Our proposed subdomain POD-TPWL (SD POD-TPWL) can be easily generalized to the global domain POD-TPWL (GD POD-TPWL). The differences between SD POD-TPWL and GD POD-TPWL are (a) model order reduction in global domain versus reduction in each subdomain separately and (b) derivative estimation using RBF interpolation in the global domain versus interpolation for each subdomain. As shown in Table 4, the total dimension of the reduced-order linear model is 18+122+51=191 for domain decomposition and 18+72+36 =126 for the global domain. Table 6 shows the total number of the reduced variables in each subdomain and in the global domain. While the total sum of the reduced variables in each subdomain is larger than that of the global domain, the number of reduced variables in each individual subdomain is relatively small. Furthermore, these local reduced variables have the surprisingly abilities to accurately capture the flow dynamics, as suggested by Fig. 13. Figure 13 shows the distribution of pressure and saturation at the final time. In this case, the reconstructions of the saturation and pressure field using a small number of patterns in each subdomain are comparable with those of the global domain. In addition, as shown in Table 7, both GD POD-TPWL and SD POD-TPWL can converge to satisfactory results. The SD POD-TPWL needs 55 FOM simulations, while the GD POD-TPWL algorithm requires 73 FOM simulations (15 FOM simulations are run to collect the snapshots, 55 FOM simulations are used to construct the initial reduced-order linear model in the first outer-loop, and the remaining 3 FOM simulations are used to reconstruct the reduced-order linear models in the following three outer-loops). Therefore, compared to the global RBF interpolation, the proposed local RBF interpolation technique requires only a small number of reduced variables per subdomain and is much more computationally efficient. If the dimension of the underlying model would be much larger, the GD POD-TPWL would result in a reduced-order linear model with a higher dimension and therefore more interpolation points would be required in the RBF scheme. In the SD POD-TPWL algorithm this problem is avoided since for large-scale problems the dimension of the reduced-order linear model for the subdomain does not increase significantly, we only need to activate more subdomains.

For Case 1, history matching results using GD POD-TPWL are slightly better than those from the subdomain POD-TPWL, especially for the high-permeable zone, e.g., the red area in Fig. 10. The water-front of the waterflooding process propagates forward quickly (as the blue area in Fig. 13) and therefore there are strong dynamic interactions

Fig. 16 Error analysis in terms of measurement and permeability misfits for the 20 solutions for case 1
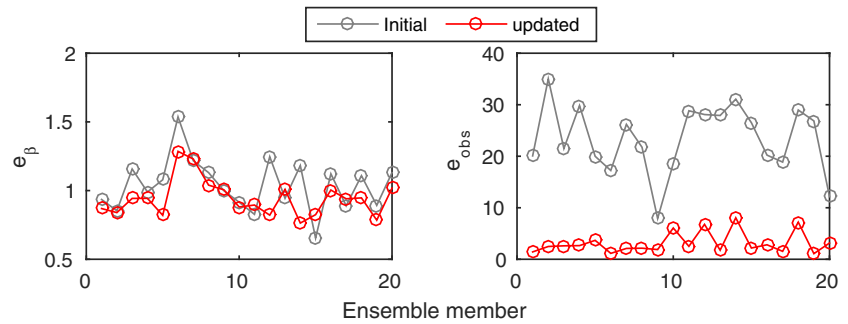
within this area. Our chosen domain decomposition may artificially cut off this inherent dynamic interaction between the east-south corner and the west-north corner. A flow-informed domain decomposition technique may therefore be required to identify the relevant dynamic interactions, especially for strongly heterogeneous reservoir models such as those based on strongly contrasting facies distributions or channels.

Solutions in our previous numerical experiments do not enable us to quantify the uncertainty of the permeability field and the predictions. In general random maximum likelihood (RML) procedure [33] enables the assessment of the uncertainty by generating multiple "samples" from the posterior distribution. Each of these samples is a history matched realization, which also honors the data. Traditional 4D-Var or gradient-based history matching method obtains only one specific solution. Additional solutions are obtained by repeatedly implementing the minimization process, but has a very highly computational cost. The RML procedure can be efficiently implemented using our proposed reduced-order history matching algorithm. When different background parameters are chosen to construct the reduced-order linear models, several valid solutions are obtained based on acceptable data misfits. In this case, we

choose 20 different background parameter sets to repeatedly implement our proposed adjoint-based reservoir history matching process.

Once the reduced model has been constructed the history matching can be efficiently repeated for different initial (background) models. Figure 14 shows an ensemble of the posterior realizations (updated log-permeability field) using 20 different background parameters sets. The main geological features, e.g., the high permeable area, are partly reconstructed in all of these 20 cases. Figures 15 and 16 show an ensemble of forecasts and the corresponding data misfits respectively using these 20 different initial and posterior models. Almost all of these 20 calibrated models produce improved predictions of the fluid rate and WCT for all eight producers that are generally consistent with the data. Thus, all of these 20 updated log-permeability fields can be regarded as acceptable solutions of the history matching problem. The spread of the predictions from the posterior realizations is significantly decreased relative to the predictions from the prior realizations. In the course of uncertainty quantification, we randomly choose these 20 background parameter sets from the prerun 52 FOM simulations which is used to construct the reduced-order linear in the first outer-loop. In addition, the analysis of

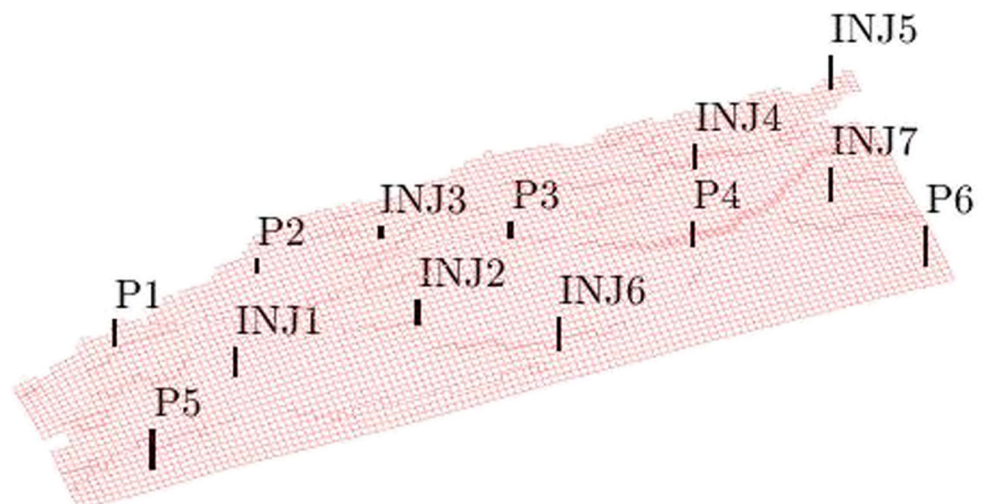**Fig. 17** The 3-D view of the well placement for case 2

Table 8 Experiment settings using MRST for case 2

Description	Value
Dimension	$40 \times 120 \times 1$
Number of wells	6 producers, 7 injectors
Constant porosity	0.2
Fluid density	1014 kg/m^3, 859 kg/m^3
Fluid viscosity	0.4 mP·s, 2 mP·s
Initial pressure	30 MPa
Initial saturation	S_o=0.80, S_w=0.20
Connate water saturation	S_{wc}=0.20
Residual oil saturation	S_{or}=0.20
Corey exponent, oil	4.0
Corey exponent, water	4.0
Injection rate	250m^3/d
BHP	25MPa
History production time	10 year
Prediction time	15 year
Timestep	0.1 year
Measurement timestep	0.2 year

the ensemble spreading enables us not to implement the additional outer-loops for updating the reduce-order linear model. This means there is no need to run additional FOM simulations for the outer-loops, which makes our method very efficient. Finally, in order to obtain these 20 solutions, the RML procedure totally requires 52 FOM simulations, including 15 FOM simulations for collecting snapshots,

(a) 'True' model in original full-order space

(b) Projected 'true' model in reduced-order subspace
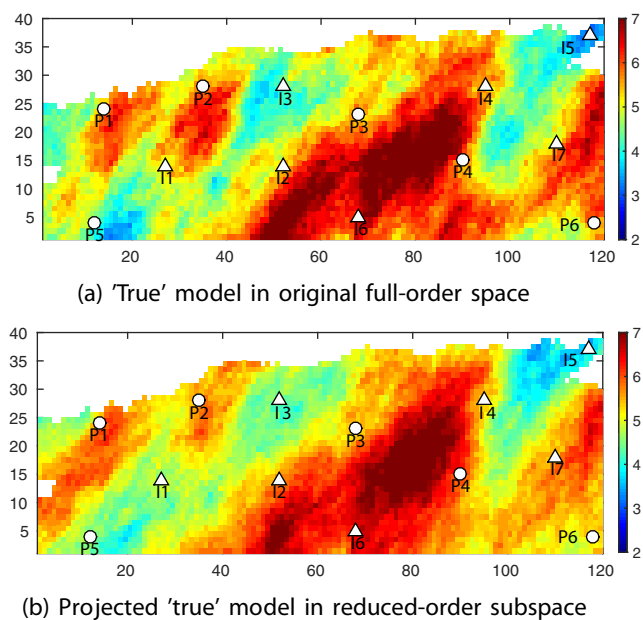
Fig. 18 The comparison of the "true" reservoir model in full-order space and reduced-order space for case 2
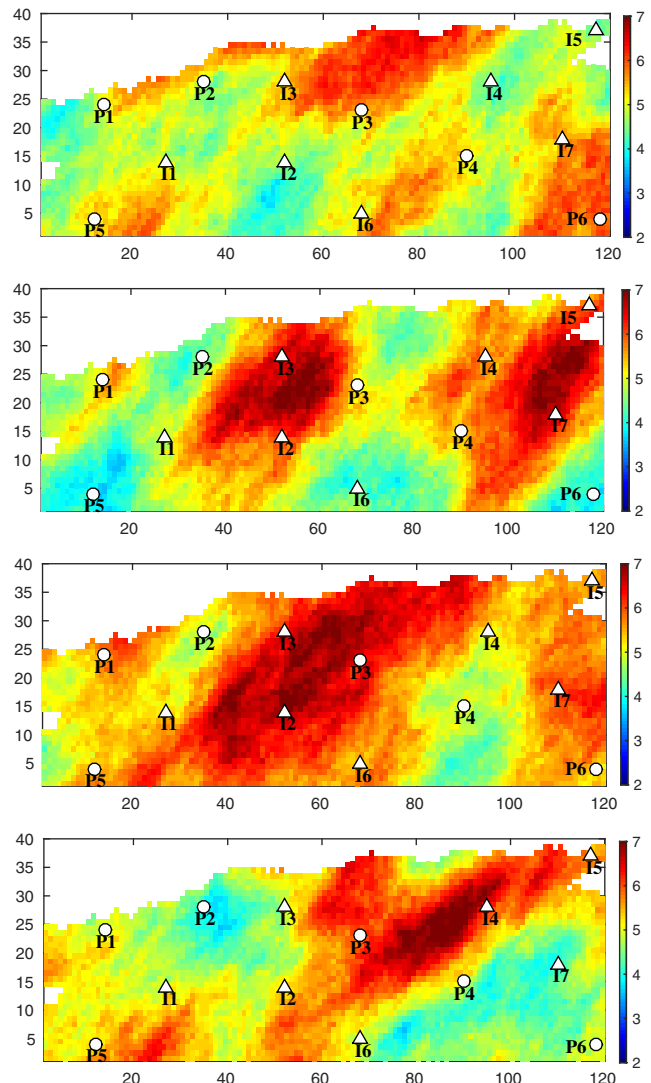
Fig. 19 Realizations of log-permeability field generated in the interval {-1, 1} for case 2

37 FOM simulations for the initial construction of the reduced-order linear model and no additional outer-loop.

5.2 Case 2 - 2D benchmark model with 13 wells

5.2.1 Description of model settings

In the second numerical experiment, the SAIGUP model [31] is used to test our proposed adjoint-based history matching approach. The first layer containing a total of 3895 active and 905 inactive grid cells is chosen for our test case. The realistic geological properties, e.g., faults, are preserved. The reservoir model describes an oil–water two-phase flow system with six producers and seven injectors, which are labeled from P_1 to P_6, and I_1 to I_7, see Fig. 17. Some detailed information about reservoir geometry, rock

Fig. 20 The 2-D view of domain decomposition in the "true" model for case 2
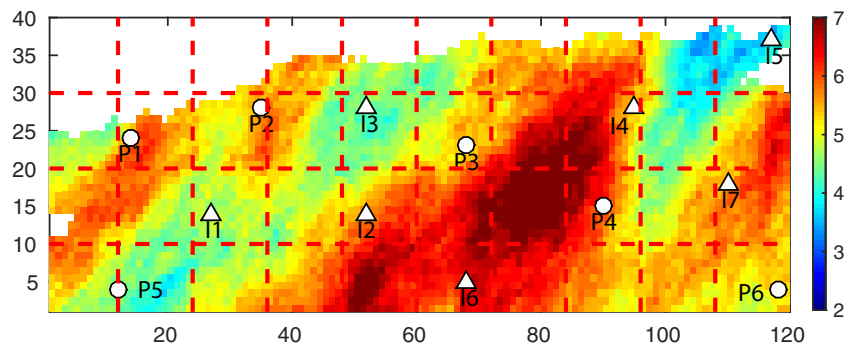
properties, fluid properties, and well controls are shown in Table 8.

5.2.2 Description of reduced model procedure

Similarly, as Eqs. 43– 46, we generate an ensemble of 1000 Gaussian-distributed realizations of log-permeability. One of these realizations was considered to be the truth, as Fig. 18a. To efficiently implement our methodology in this much more realistic case, the log-permeability field was parameterized using a KL-expansion described above and about 90% energy is maintained, resulting in 44 permeability patterns to represent the uncertainty in all 3895 active grid cells. Figure 18b represents the "true" permeability field projected onto the subspace spanned by these l_β=44 PCA coefficients. Most parts of the properties of the original "true" model is reconstructed in the reduced-order parameter subspace, while parts of properties are lost, e.g., the high-permeable area around the producer P_2. Four realizations for log-permeability field generated from random coefficients sampled from the set {-1, 1} are illustrated in Fig. 19.

Experiments showed that the changes in singular value spectrum of the snapshot matrix are insignificant when the snapshots at every timestep are selected from 20 or more FOM simulations. These 20 FOM are also sampled from the interval {-1, 1} to effectively preserve the dynamic behavior. Finally, we collect 2000 snapshots for pressures and saturations separately. Comparing to the previous synthetic model, the existence of the faults and strong heterogeneity makes the flow dynamic of SAIGUP more complicated. The error quantification for subdomain POD-TPWL is also implemented and we obtain some consistent results with the Case 1; thus, the detailed implementation of error quantification is not described here. To effectively capture the local physical features, we divide the whole model domain into 40 subdomains (4 subdomains in x direction times 10 subdomains in y direction) as in Fig. 20. The inactive grids are intentionally considered to convert the original irregular model into a rectangle reservoir model, and subsequently this regular model can be conveniently

decomposed into subdomains. In addition, to increase the efficiency, we do not construct the reduced-order linear model in the subdomain if the number of active grids is less than 5. For each subdomain, two eigenvalue problems separately for pressure and saturation are solved using POD. Both decompositions preserve 90% of the energy, and the number of reduced variables for each subdomain is shown in Fig. 21.

The history period is 10 years during which observations are taken from these 13 wells every 0.2 years, resulting in 300 fluid rates and 300 WCT values measured in the producers and 350 bottom-hole pressures measured in the injectors, which give in total 950 data points. The generated measurements for producing wells are shown in Fig. 22.

5.2.3 History matching results

The accuracy of RBF interpolation depends on the interpolation points, and the number of interpolation points depends on the number of interpolation variables. In the
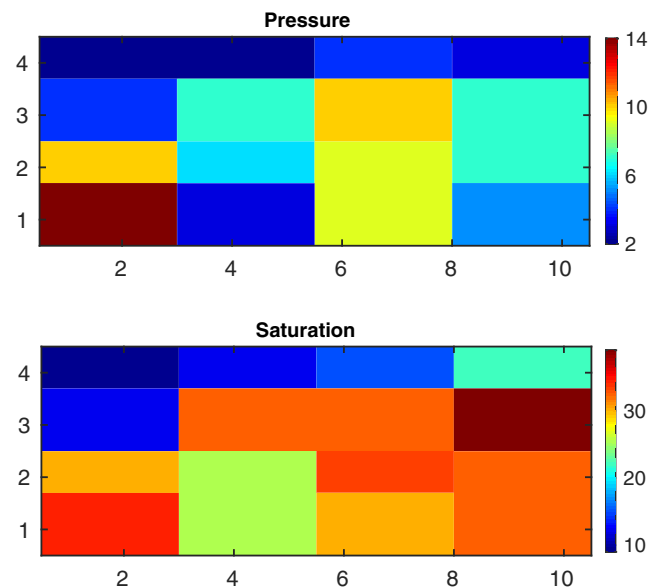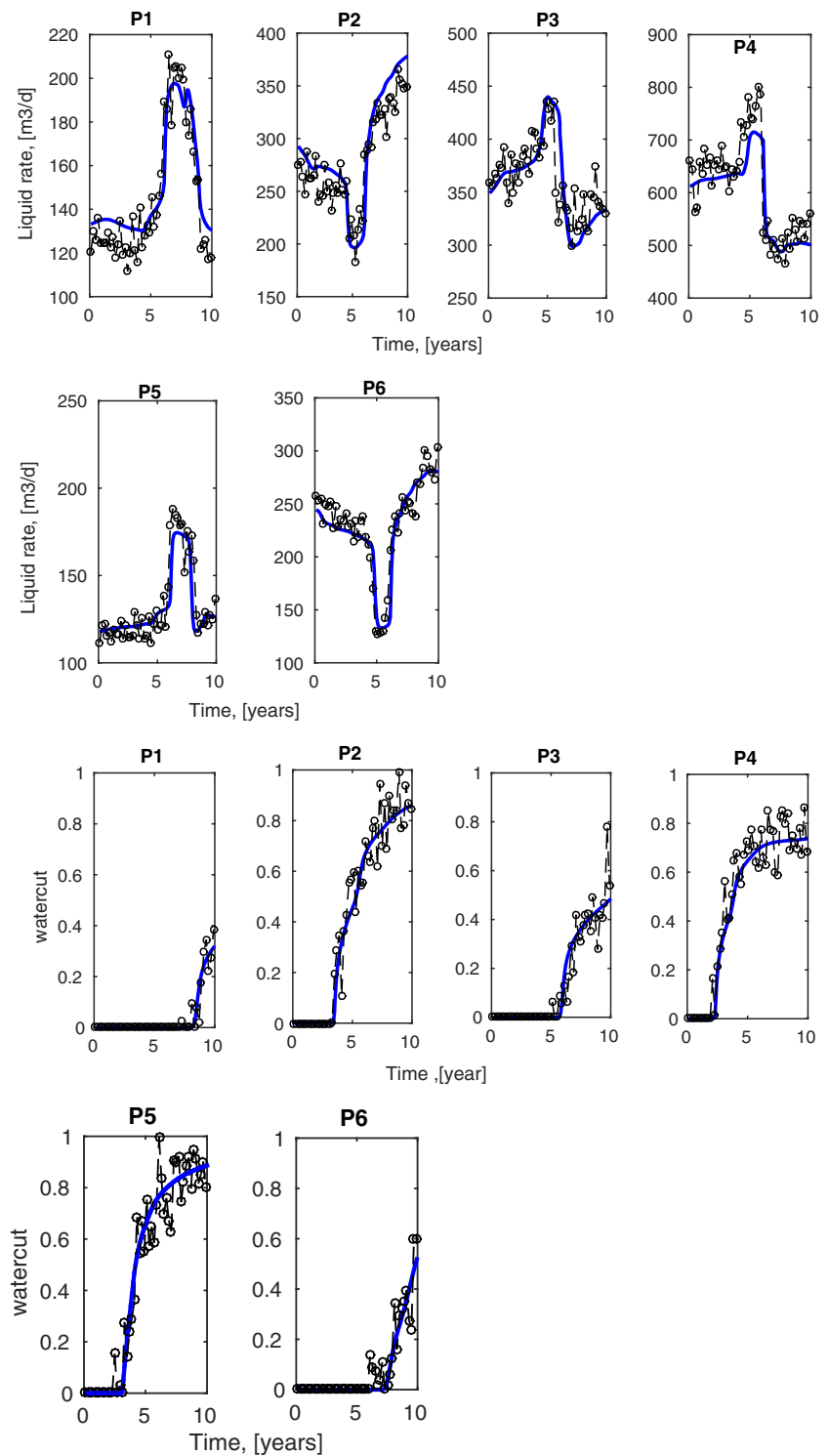

Fig. 21 The number of reduced pressure and saturation variables in each subdomain for case 2

Fig. 22 Example of synthetic measurements for case 2: green line-initial model, blue line-reference model, dot line-noisy data

previous synthetic model (retaining 18 PCA coefficients), the number of sampling points using the simple two-sides perturbation strategy is sufficient to represent the interval {-1, 1}. However, in this case, retaining 44 PCA coefficients forces us to sample much more interpolation points. We also compare the results with those of the FD

based history matching algorithm, which is implemented in the reduced-order parameter subspace spanned by the 44 log-permeability patterns. The one-side FD method is used to compute the numerical gradient of cost function as Eq. 51 with respect to 44 PCA coefficients. Thus, 45 FOM simulations are required for each iteration step. The

(a) 'True' Permeability

(b) Initial Permeability

(c) Updated Permeability using $2 \times l_\beta$

(d) Updated Permeability using $2 \times l_\beta$ via continuing iterations

(e) Updated Permeability using $3 \times l_\beta$

(f) Updated Permeability using $4 \times l_\beta$

(g) Updated Permeability using finite-difference method
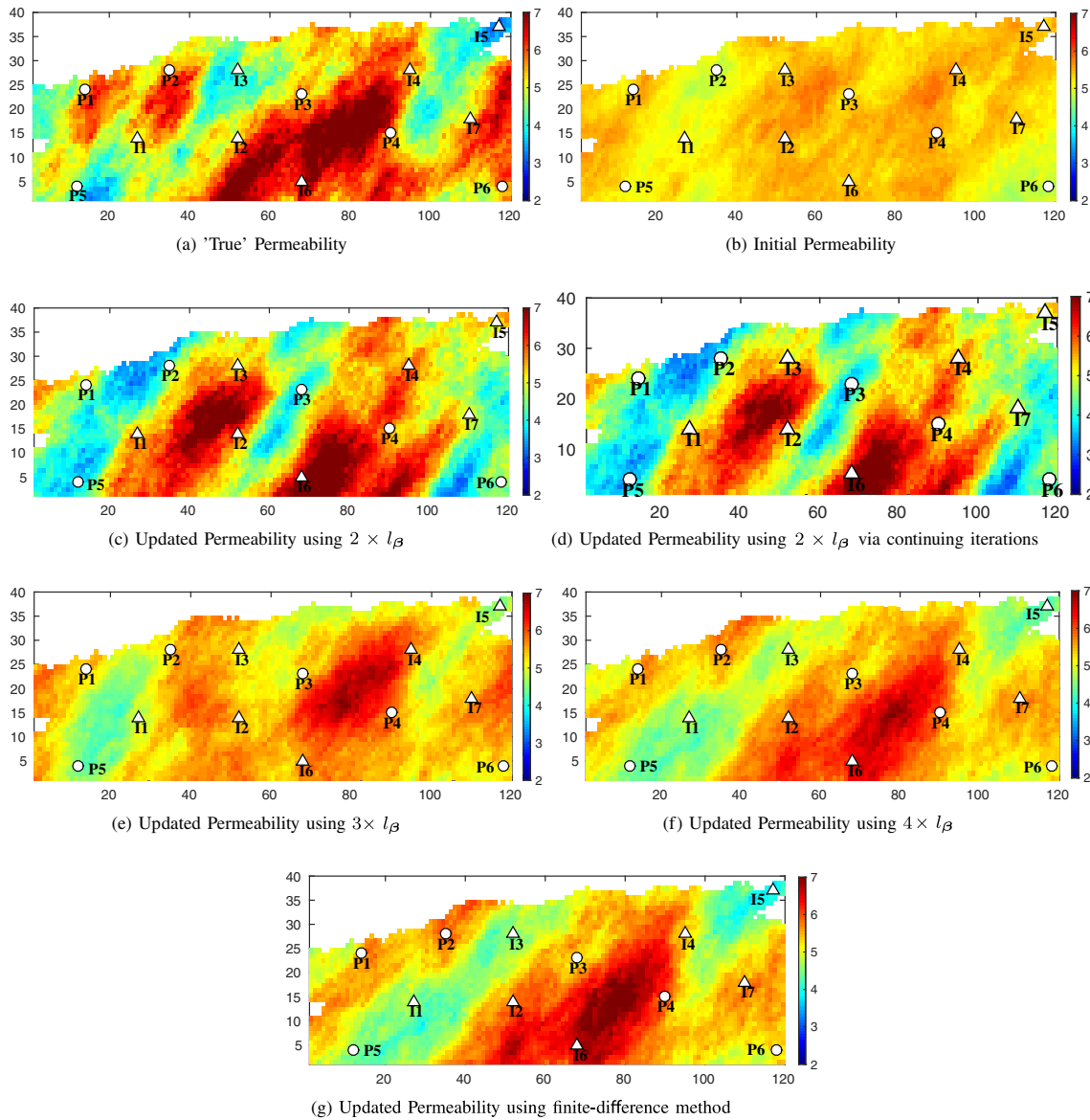
Fig. 23 The updated permeability under different number of sampling points for case 2. The stopping criteria for (**c**), (**e**), (**f**) and (**g**) is $\eta_J = 10^{-4}$, $\eta_\xi = 10^{-3}$, N_{max}=50, while The stopping criteria for (**d**) is $\eta_J = 10^{-5}$, $\eta_\xi = 10^{-4}$, N_{max}=50

stopping criteria are set to $\eta_J = 10^{-4}$, $\eta_\xi = 10^{-3}$, N_{max}=50 in this case.

Figures 23 and 24 and Table 9 show the updated log-permeability field using different numbers of sampling points. In our case, the sampling strategy is that the first $2 \times l_\beta$ sampling points are selected from the set {-1,1} through perturbing each PCA coefficient sequentially in 2 opposite directions (positive and negative), and then the remaining sampling points are chosen randomly within the interval [-1, 1]. During the minimization procedure, a small iteration step is used to ensure a decreasing cost function when we use a small number of sampling points, e.g., $2 \times l_\beta$, which leads to a less accurate approximate gradient.

Figure 24 shows that using $2 \times l_\beta$ sampling points leads to relatively slow convergence. In our predefined stopping criteria, totally 221 iteration steps with five outer-loops are required. Figure 23 illustrates that the "true" log-permeability field is approximately calibrated when increasing the number of sampling points, e.g., $3 \times l_\beta$ and even $4 \times l_\beta$ in our case. It is easily recognized that using small number of sampling points leads to bad quality of approximate gradients derived from the subdomain reduced adjoint model, which will deteriorate the minimization procedure. We continue the iteration process when using $2 \times l_\beta$ sampling points, to investigate whether there are potentials to obtain comparable results with that of FD
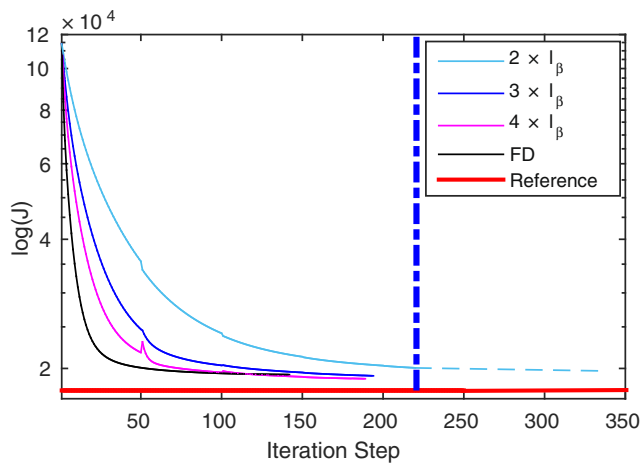
Fig. 24 Comparison of the cost function value using different number of sampling points for case 2. The extended dash line represents the cost function using a new stopping criterion $\eta_J = 10^{-5}$, $\eta_\xi = 10^{-4}$, N_{max}=50

Fig. 25 Realizations of the log-permeability field generated in the interval {-0.1, 0.1} for case 2

method. As Fig. 24, the vertical blue dash line represents the starting point using new stopping criterion, and the dash line represents the corresponding cost function of the new iteration process. Another 122 iteration steps and 2 outer-loops are required to reach convergence. In contrast to the old stopping criterion, the final cost function is decreased by 4.9% in this case. We should note that these additional 122 iteration steps just require 2 new FOM simulations for the two outer-loops. This property makes our methodology significantly attractive because much more iteration steps do not explosively increase the FOM simulations. Comparison between Fig. 23c, d also shows that the log-permeability filed has slight improvement when continuing the iteration process. Comparison between Fig. 23f, g demonstrates that our model-reduced approach obtains comparable updated log-permeability field with the FD method using $4 \times l_\beta$ sampling points, e.g., 199 FOM simulations, and among them, 20 FOM simulations are run to collect the snapshots, 176 FOM simulations are used to construct the initial
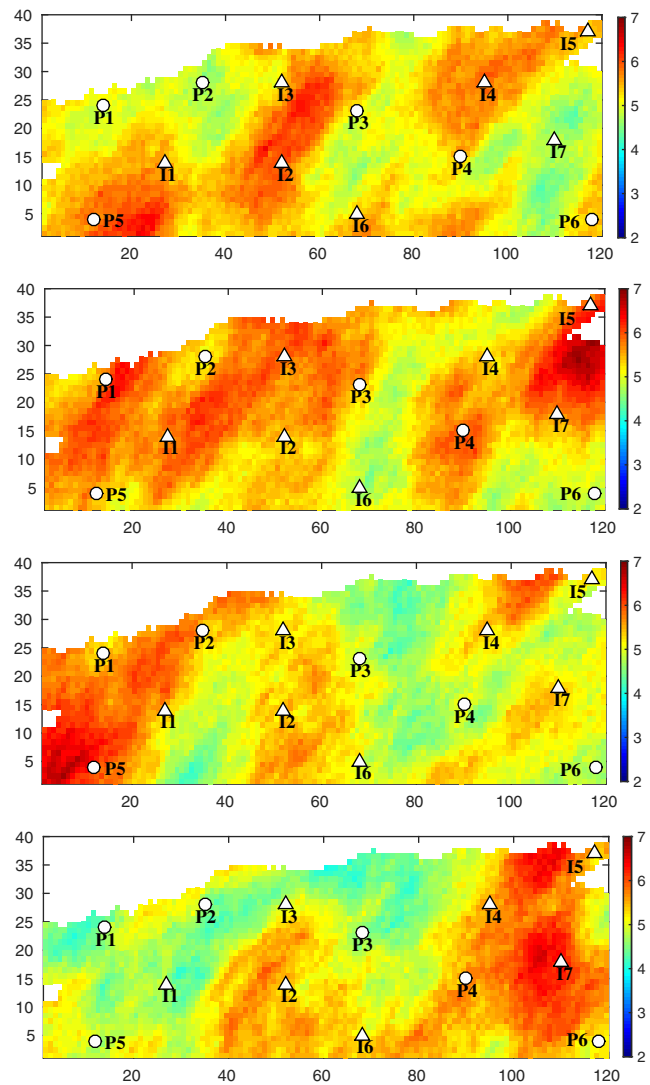
reduced-order linear model in the first outer-loop, the remaining 3 FOM simulations are used to reconstruct the reduced-order linear models in the next three outer-loops

Table 9 Comparison between SD POD-TPWL and FD method using different number of sampling points for case 2

–	Iterations	FOM	$J(\xi) \times 10^5$	e_{obs}	e_β
Initial model	–	–	1.1417	73.5892	0.7078
SD POD-TPWL[1] ×2	221	113	0.2117	18.1208	0.7837
SD POD-TPWL[2] ×2	343	115	0.1998	16.3524	0.7797
SD POD-TPWL×3	194	155	0.1931	14.8523	0.5017
SD POD-TPWL×4	189	199	0.1892	11.0852	0.4182
FD	78	3510	0.1935	11.2532	0.3925
"True" model	–	–	0.1777	9.2531	0

SD POD-TPWL[1] ×2 means the stopping criteria $\eta_J = 10^{-4}$, $\eta_\xi = 10^{-3}$, N_{max}=50, while SD POD-TPWL[2] ×2 means the stopping criteria $\eta_J = 10^{-5}$, $\eta_\xi = 10^{-4}$, N_{max}=50
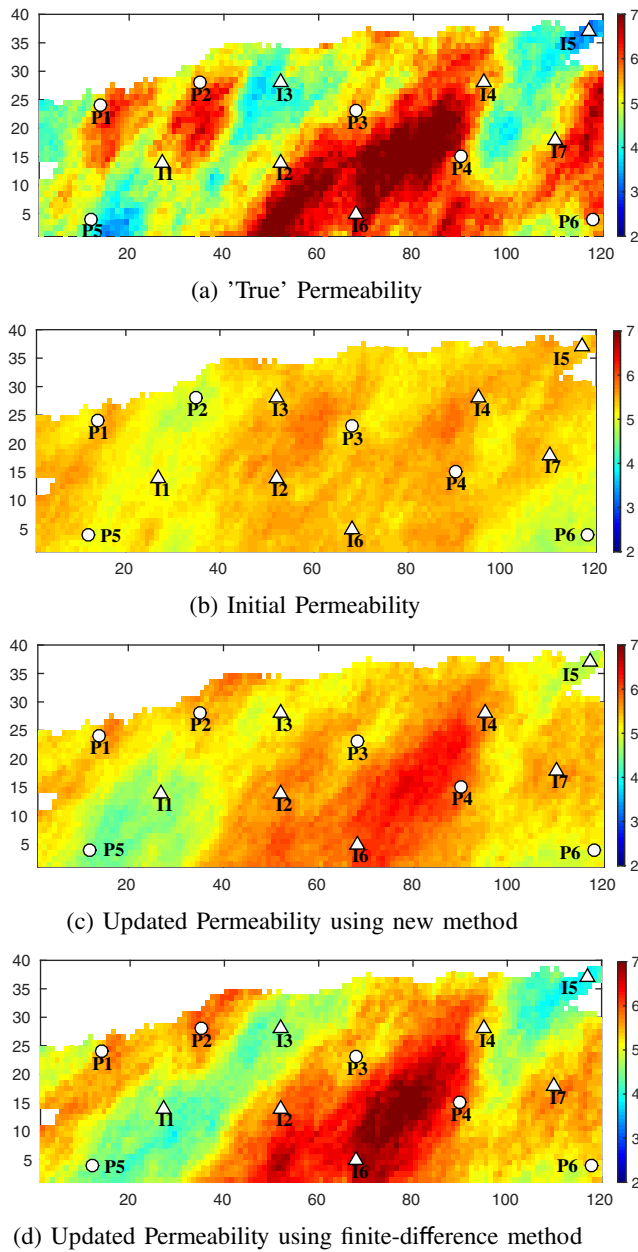
(a) 'True' Permeability

(b) Initial Permeability

(c) Updated Permeability using new method

(d) Updated Permeability using finite-difference method

Fig. 26 Comparison of the initial, "true" and updated log-permeability using interval {-0.1, 0.1} for case 2

Fig. 27 Cost function value of SD POD-TPWL and FD method using interval {-0.1, 0.1} for case 2

and calculate the cost function as Eq. 3 in original space, while the FD method requires 3510 FOM simulations.

Alternatively, we limit the sampling points within a relative small interval {-0.1, 0.1} so that only a small number of sampling points, e.g., two-side perturbations (2×44+1=89), is required to efficiently construct a subdomain reduced-order linear model. Four realizations for log-permeability field generated from this small interval {-0.1, 0.1} are illustrated in Fig. 25. Figures 26, 27, and 28 and Table 10 show the results of these numerical experiments, including the updated log-permeability field, iterative value of cost

function and the mismatch between observed data and predictions. As Fig. 28 and Table 10, both SD POD-TPWL and FD method are able to converge to a satisfactory minimum cost function value. FD method has relatively high convergence ratio over our SD POD-TPWL, e.g., 78 and 195 iteration steps with 4 outer-loops separately, and also yields slightly more accurate updated log-permeability. However, our model-reduced approach only needs 112 FOM simulations, while the FD method requires 3510 FOM simulations. Both PCA-based parameterization and POD-based model reduction introduce some inherent approximation errors into this minimization procedure; therefore, the SD POD-TPWL and the FD method in this case generally do not decrease the cost function to the reference value illustrated as the bold red line. In addition, POD-based model reduction also makes our method theoretically less accurate than that of the FD method, which is demonstrated as the black line and blue line in the Fig. 26.

There is a trade-off between the number of sampling points and the parameter interval. On the one hand, a large parameter interval generally has high possibilities to include the parameter space containing the "true" model; thus, the "true" solution is likely to be found while needs to use a large number of sampling points. On the other hand, we also can choose a small number of sampling points within a small interval. Although there is a larger possibility that the "true" solution is not contained in this interval, a valid solution (based on an acceptable data mismatch) can most likely be found using a relatively small number of sampling points. Therefore, it is highly significant to choose reasonable parameter interval in practice. If we have poor prior information for the "true" model, a relative large parameter interval is preferable to provide relatively more
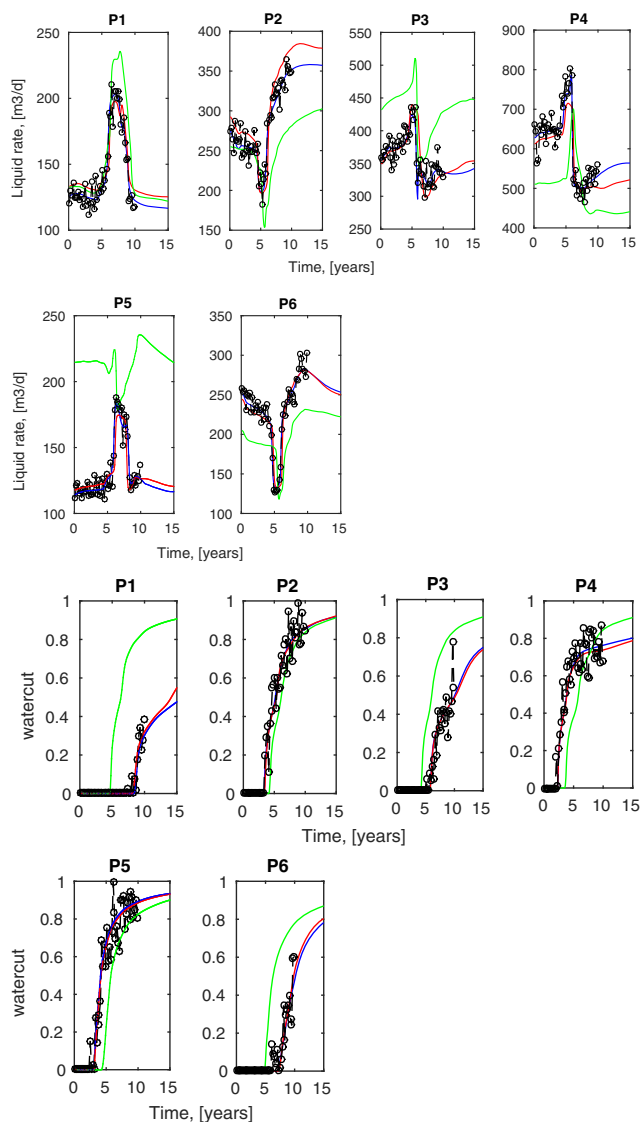
Fig. 28 Forecast of the liquid rate and water-cut using interval {-0.1, 0.1} for case 2: green line-initial model, blue line-reference model, red line-updated model by subdomain POD-TPWL

Table 10 Comparison of subdomain POD-TPWL and FD method using interval {-0.1, 0.1} for case 2

–	Iterations	FOM	$J(\boldsymbol{\xi}) \times 10^5$	e_{obs}	e_β
Initial model	–	–	1.1417	73.5892	0.7078
SD POD-TPWL	195	112	0.1928	11.1802	0.4685
FD	78	3510	0.1935	11.2532	0.3925
"True" model	–	–	0.1777	9.2531	0

eigenvalue decomposition of the covariance matrix, (2) implementing model order reduction using POD in each subdomian, (3) conducting RBF interpolation and computing the derivative matrices. The cost of eigenvalue decomposition and POD is negligible for small models, while it will become significant for large-scale models. In our cases, the required number of FOM simulations is roughly 2-3 times the number of PCA coefficients, e.g., 54 simulations for the synthetic model, 113 (sampling within a small interval [-0.1, 0.1]) and 199 (sampling within a relative large interval [-1,1]) FOM simulations for the SAIGUP model, respectively. This process is code non-intrusive without the need of large programming effort. Besides, this process is also easily parallelized. Once available, the costs of running the reduced model are negligible. We should note that the gradient-based reservoir history matching generally requires $O(10^2 - 10^4)$ FOM simulations, thus, an offline cost of $O(10 - 10^2)$ FOM simulations in these settings is attractive. For large-scale reservoir history matching, the main computational cost is dominated by the required number of FOM simulations. In our proposed method, most part of the FOM simulations is mainly in offline stage, which means that our method is easily implemented.

accurate results, e.g., finding the "true" solution, of history matching process, as a compensation, a large number of sampling points are required to implement our SD POD-TPWL. On the contrary, if we have good prior information for the "true" model, a small parameter interval can be perturbed around the prior parameter field. Using a small number of sampling points enables us to obtain satisfactory history matching results.

5.3 Computational aspects

This section discusses the computational aspects of our proposed adjoint-based reservoir history matching algorithm. The offline computational costs for subdomain POD-TPWL algorithm comprise (1) executing parameterization using

6 Conclusions

We have introduced a variational data assimilation method where the adjoint model of the original high-dimensional non-linear model is replaced by a subdomain reduced-order linear model. Parameterization and proper orthogonal decomposition techniques are used to simultaneously reduce the parameter space and reservoir model. In order to avoid the need for simulator code access and modification and numerous full-order model simulations, we integrated domain decomposition and radial basis function interpolation with trajectory piecewise linearization to form a new subdomain POD-TPWL algorithm. The reduced-order linear model is easily incorporated into an adjoint-based parameter estimation procedure. The use of domain decomposition allows for large-scale applications since the number of interpolation points required depends primarily on the

number of the parameters and not on the dimension of the underlying full-order model.

We used the subdomain model-reduced adjoint-based history matching approach to calibrate the unknown permeability fields of a small 2D synthetic model and of a modified version of the SAIGUP benchmark model with noisy synthetic measurements. The permeability field is parameterized using a KL-expansion resulting in a small number of permeability patterns that are used to represent the original gridblock permeability. The reservoir domains for the small model and the SAIGUP model are divided into 9 and 40 subdomains respectively. In the first numerical experiment, our methodology accurately reconstructs the "true" permeability field and shows similar results as more classic finite-difference based history matching. Our method also significantly improves the prediction of fluid rate and water breakthrough time of production wells. Without any additional full-order model simulations, our approach efficiently generates an ensemble of models that all approximately match the observations. The results of the second example show that also for a more complex history matching problem, very promising results could be obtained. For the cases studied in this paper, the number of full-order model simulations required for history matching is roughly 2–3 times the number of the number of global parameter patterns.

There are a number of aspects of the proposed methodology that could possibly be improved. In example 2, it was observed that sampling strategy has to be chosen with care to obtain an efficient implementation. Some diagnostics could possibly be devised to determine if and how many additional sampling points need to be generated. We have chosen somewhat arbitrary decompositions of the global domain into subdomains. In the first example it was observed that it may be beneficial to choose the subdomains based on information about the main dynamical patterns. Since in reservoir applications these patterns are strongly affected by the placement of producers and injectors the subdomain decomposition could possibly be informed by the well lay-out. In this paper, we considered a global parameterization of the log-permeability field where the PCA patterns are defined over the entire domain. From a computational point of view, a local parameterization where the parameters are defined in each subdomain separately is very attractive. Since in this case parameters can be perturbed independent of each other and the effects of all these perturbations can be computed with very few full-order model simulations. The local parameterization technique is the focus of our ongoing research.

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

References

1. Altaf, M.U., Heemink, A.W., Verlaan, M.: Inverse shallow-water flow modeling using model reduction . Int. J. Multiscale Comput. Eng. **7**(6), 577–594 (2009)
2. Amsallem, D., Zahr, M.J., Farhat, C.: Nonlinear model order reduction based on local reduced-order bases. Int. J. Numer. Methods Eng. **92**(10), 891–916 (2012)
3. Baiges, J., Codina, R., Idelsohn, S.: A domain decomposition strategy for reduced order models. application to the incompressible navier–stokes equations. Comput. Methods Appl. Mech. Eng. **267**, 23–42 (2013)
4. Bian, X., Li, Z., Karniadakis, G.E.: Multi-resolution flow simulations by smoothed particle hydrodynamics via domain decomposition. J. Comput. Phys. **297**, 132–155 (2015)
5. Bishop, C.H., Frolov, S., Allen, D.R., Kuhl, D.D., Hoppel, K.: The local ensemble tangent linear model: an enabler for coupled model 4d-var. Q. J. Roy. Meteorol. Soc. **143**(703), 1009–1020 (2017)
6. Bruyelle, J., Guérillot, D.: Neural networks and their derivatives for history matching and reservoir optimization problems. Comput. Geosci. **18**(3-4), 549 (2014)
7. Cardoso, M., Durlofsky, L., Sarma, P.: Development and application of reduced-order modeling procedures for subsurface flow simulation. Int. J. Numer. Methods Eng. **77**(9), 1322–1350 (2009)
8. Cardoso, M., Durlofsky, L.J.: Linearized reduced-order models for subsurface flow simulation. J. Comput. Phys. **229**(3), 681–700 (2010)
9. Chaturantabut, S.: Temporal localized nonlinear model reduction with a priori error estimate. Appl. Numer. Math. **119**, 225–238 (2017)
10. Chinchapatnam, P.P., Djidjeli, K., Nair, P.B.: Domain decomposition for time-dependent problems using radial based meshless methods. Numer. Methods Partial Differential Equations **23**(1), 38–59 (2007)
11. Courant, R., Hilbert, D.: Methods of mathematical physics: Wiley interscience (1962)
12. Courtier, P., Thépaut, J.N., Hollingsworth, A.: A strategy for operational implementation of 4d-var, using an incremental approach. Q. J. Roy. Meteorol. Soc. **120**(519), 1367–1387 (1994)
13. Evans, D.J.: Parallel sor iterative methods. Parallel Comput. **1**(1), 3–18 (1984)
14. Frolov, S., Bishop, C.H.: Localized ensemble-based tangent linear models and their use in propagating hybrid error covariance models. Mon. Weather. Rev. **144**(4), 1383–1405 (2016)
15. Fukunaga, K., Koontz, W.L.: Application of the Karhunen-Loeve expansion to feature selection and ordering. IEEE Trans. Comput. **100**(4), 311–318 (1970)

16. Golub, G., Ortega, J.M.: Scientific computing: an introduction with parallel computing . Academic Press, Cambridge (1993)

17. He, J., Durlofsky, L.J.: Constraint reduction procedures for reduced-order subsurface flow models based on pod-tpwl. Int. J. Numer. Methods Eng. **103**(1), 1–30 (2015)

18. He, J., Durlofsky, L.J., et al.: Reduced-order modeling for compositional simulation by use of trajectory piecewise linearization. SPE J. **19**(05), 858–872 (2014)

19. He, J., Sætrom, J., Durlofsky, L.J.: Enhanced linearized reduced-order models for subsurface flow simulation. J. Comput. Phys. **230**(23), 8313–8341 (2011)

20. He, J., Sarma, P., Durlofsky, L.J.: Reduced-order flow modeling and geological parameterization for ensemble-based data assimilation. Comput. Geosci. **55**, 54–69 (2013)

21. Heijn, T., Markovinovic, R., Jansen, J., et al.: Generation of low-order reservoir models using system-theoretical concepts. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers (2003)

22. Jansen, J.: Adjoint-based optimization of multi-phase flow through porous media-a review. Comput. Fluids **46**(1), 40–51 (2011)

23. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. J. Glob. Optim. **21**(4), 345–383 (2001). https://doi.org/10.1023/A:1012771025575

24. Kaleta, M.P., Hanea, R.G., Heemink, A.W., Jansen, J.D.: Model-reduced gradient-based history matching. Comput. Geosci. **15**(1), 135–153 (2011)

25. Klie, H. et al.: Unlocking fast reservoir predictions via non-intrusive reduced-order models. In: SPE reservoir simulation symposium. Society of Petroleum Engineers (2013)

26. Lie, K.A., Krogstad, S., Ligaarden, I.S., Natvig, J.R., Nilsen, H.M., Skaflestad, B.: Open-source matlab implementation of consistent discretisations on complex grids. Comput. Geosci. **16**(2), 297–322 (2012)

27. Liu, C., Xiao, Q., Wang, B.: An ensemble-based four-dimensional variational data assimilation scheme. part i: Technical formulation and preliminary test. Mon. Weather. Rev. **136**(9), 3363–3373 (2008)

28. Liu, C., Xiao, Q., Wang, B.: An ensemble-based four-dimensional variational data assimilation scheme. part ii: Observing system simulation experiments with advanced research wrf (arw). Mon. Weather. Rev. **137**(5), 1687–1704 (2009)

29. Lucia, D.J., King, P.I., Beran, P.S.: Reduced order modeling of a two-dimensional flow with moving shocks. Comput. Fluids **32**(7), 917–938 (2003)

30. Markovinović, R., Jansen, J.: Accelerating iterative solution methods using reduced-order models as solution predictors. Int. J. Numer. Methods Eng. **68**(5), 525–541 (2006)

31. Matthews, J.D., Carter, J.N., Stephen, K.D., Zimmerman, R.W., Skorstad, A., Manzocchi, T., Howell, J.A.: Assessing the effect of geological uncertainty on recovery estimates in shallow-marine reservoirs: the application of reservoir engineering to the saigup project. Pet. Geosci. **14**(1), 35–44 (2008). https://doi.org/10.1144/1354-079307-791. http://pg.lyellcollection.org/content/14/1/35

32. Nocedal, J., Wright, S.J.: Numerical optimization: Springer (1999)

33. Oliver, D.S., Reynolds, A.C., Liu, N.: Inverse theory for petroleum reservoir characterization and history matching. Cambridge University Press, Cambridge (2008)

34. Peaceman, D.W.: Fundamentals of Numerical Reservoir Simulation. Elsevier Scientific Publishing Company, Amsterdam (1977)

35. Przemieniecki, J.S.: Matrix structural analysis of substructures. AIAA J. **1**(1), 138–147 (1963)

36. Smolyak, S.: Quadrature and interpolation formulas for tensor products of certain classes of functions. Soviet Math. Dokl. **4**, 240–243 (1963)

37. Tan, X., Gildin, E., Florez, H., Trehan, S., Yang, Y., Hoda, N.: Trajectory-based DEIM (TDEIM) model reduction applied to reservoir simulation. Comput. Geosci. https://doi.org/10.1007/s10596-018-9782-0 (2018)

38. Tarantola, A.: Inverse problem theory: Methods for data fitting and model parameter estimation. DBLP (2005)

39. Trehan, S., Durlofsky, L.J.: Trajectory piecewise quadratic reduced-order model for subsurface flow, with application to pde-constrained optimization. J. Comput. Phys. **326**, 446–473 (2016)

40. Vermeulen, P., Heemink, A.: Model-reduced variational data assimilation. Mon. Weather. Rev. **134**(10), 2888–2899 (2006)

41. Wu, Y., Wang, H., Zhang, B., Du, K.L.: Using radial basis function networks for function approximation and classification. ISRN Applied Mathematics **2012**, 34 (2012)

42. Xiao, D., Fang, F., Pain, C., Navon, I., Muggeridge, A.: Non-intrusive reduced order modelling of waterflooding in geologically heterogeneous reservoirs. In: ECMOR XV-15th European conference on the mathematics of oil recovery (2016)

43. Xiao, D., Fang, F., Pain, C., Navon, I., Salinas, P., Muggeridge, A.: Non-intrusive model reduction for a 3d unstructured mesh control volume finite element reservoir model and its application to fluvial channels. Computers & Geosciences (2016)

Affiliations

Cong Xiao[1] · Olwijn Leeuwenburgh[2,3] · Hai Xiang Lin[1] · Arnold Heemink[1]

Olwijn Leeuwenburgh
olwijn.leeuwenburgh@tno.nl

Hai Xiang Lin
h.x.lin@tudelft.nl

Arnold Heemink
A.W.Heemink@tudelft.nl

[1] Delft Institute of Applied Mathematics, Delft University
of Technology, Van Mourik Broekmanweg 6, 2628 XE Delft, The
Netherlands

[2] Civil Engineering and Geosciences, Delft University
of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

[3] TNO, Princetonlaan 6 PO Box 80015 3508 TA Utrecht,
The Netherlands