A multiscale method for large-scale inverse modeling: single-phase transient flow

Jianlin Fu *  
Jef Caers †  
Hamdi Tchelepi ‡

Department of Energy Resources Engineering  
Stanford University  
367 Panama Street, Stanford, CA 94305, USA  
Tel. 650 723 8064  
Fax. 650 725 2099

Abstract
High-resolution geologic models that incorporate observed state data are expected to effectively enhance the reliability of performance prediction. One of the major challenges faced is how to solve the large-scale inverse modeling problem, i.e., to infer high-resolution models from the given observed state data that are related to the model parameters according to the known physical rules, e.g., the flow and transport partial differential equations. There are basically two problems, one is the high-dimensional problem and the other is the inverse problem. A multiscale method is presented in this work to attack these problems with the aid of a gradient-based optimization algorithm. Within this method, the model responses (i.e., the simulated state data) are very efficiently computed from the high-resolution model using the multiscale finite-volume method. The mismatch between the observations and the simulations is then defined as the objective function and the fine-scale sensitivity coefficient (i.e., the derivative of the objective function with respect to each node’s attribute) is computed by a multiscale adjoint method for subsequent optimization. The difficult high-dimensional optimization problem is reduced to a one-dimension using the gradient-based gradual deformation method. A synthetic single-phase transient flow example problem is employed to

*Corresponding author. E-mail: fu.jianlin_ac@yahoo.com  
†E-mail: jcaers@stanford.edu  
‡E-mail: tchelepi@stanford.edu
illustrate the efficiency of the proposed method. Results demonstrate that the multiscale framework presented is not only computationally efficient but also can generate geologically consistent models. The prior information preservation of this method overcomes the artifacts introduced by the multiscale simulation and significantly enhances the prediction ability of the inverse-conditional realization generated.

1 Introduction

The reservoir performance is largely determined by the heterogeneity of the porous media. Current geological and geostatistical modeling tools allow us to generate very high-dimensional models that are capable of sufficiently capturing the detailed spatial variability of petrophysical attributes, e.g., permeability. Traditional reservoir flow simulation methods unfortunately do not keep up with this advance in reservoir characterization or reservoir modeling, which causes to a significant scale discrepancy between the reservoir modeling grid (of order $10^6$) and the flow simulation grid (of order $10^5$). (For the convenience of presentation, we use “modeling” to refer to geologic or reservoir modeling and “simulation” to refer to reservoir flow simulation.) The case is even much worse if one tries to incorporate the state observations into the guessed models in order to reduce the model uncertainty and improve reservoir performance prediction. This is because a notoriously ill-posed, large-scale inverse problem arises, which is also known as model calibration in hydrogeology or history matching in petroleum engineering. This inverse modeling problem is often cast as an optimization problem (e.g., Carrera and Neuman, 1986; Datta-Gupta et al., 1995): an objective function is first defined by the mismatch between the observations and the simulations computed from an initial guessed model, then the objective function is minimized by tuning the initial model parameters using some optimization subroutines (either stochastic or gradient-based). During this procedure, the forward simulation is repeatedly run until a desired mismatch is obtained. One can easily see the major difficulties of this large-scale inverse modeling problem: one is the inverse problem and the other is the high-dimensional problem. Moreover, the geological information should be honored for the inverse modeling since the geological constraints essentially control the flow and transport of fluid. The last decades saw a few advances in addressing the inverse problem while maintaining the geological consistence, e.g., the sequential self-calibration method (Gómez-Hernández et al., 1997), the gradual deformation method (Hu, 2000), the probability perturbation method (Caers and Hoffman, 2006), and the block Markov chain Monte Carlo method (Fu and Gómez-Hernández, 2009).

A larger challenge emerges when the high-dimensional problem is accounted for. One of the most typical ways to attack this problem is the upscaling-downscaling method that tries to reduce the dimension of the problem to one that traditional reservoir simulators can accommodate (e.g., Tran et al., 2001; Yoon et al., 2001; Aanonsen, 2008). There are two alternatives in this method. One option is to apply the history matching on the coarse-scale models. The
matched coarse-scale models are then stochastically downscaled to the fine scale as required. Various mathematical methods have been developed and applied to the downscaling procedure, such as the sequential simulation with block kriging and Markov random model. In fact, the downscaled fine-scale model might produce rather large deviation from the real, fully fine-scale flow simulation responses. The other option works directly on the fine-scale model. The fine-scale model is first upscaled to the coarse scale such that the forward simulators can afford. Then the coarse-scale mismatch is defined as the objective function for fine-scale model perturbation. This procedure is repeated until the objective function is minimized. It should be pointed out that both options are identical since history matching is essentially performed at the coarse scale due to the infeasibility of the fine-scale flow simulation.

As a more robust alternative, the multiscale method has been proposed to solve the large-scale flow equation (Hou and Wu, 1997; Jenny et al., 2003). The multiscale finite-volume method (MSFV) developed by Jenny et al. (2003, 2004, 2006) is an efficient, approximate fine-scale flow simulator that offers a tool to solve the high-resolution flow problems at an inexpensive cost. Unlike traditional upscaling methods, the MSFV method is not only capable of capturing the large-scale trends of the system but also treats the problem at the small scale. This is accomplished by partitioning the original fine-scale problem into a set of primal coarse blocks with a suite of auxiliary dual volumes. A set of basis functions are built for each dual volume, which serve to assemble the coarse-scale transmissibilities for the primal flow simulation and to interpolate the coarse-scale pressures for the fine-scale reconstruction. A large number of case studies (e.g., Lunati and Jenny, 2006; Tchelepi et al., 2007; Lee et al., 2008) proved that this method is accurate, efficient, and robust for highly heterogeneous reservoir models.

On the basis of the MSFV formulation, Fu et al. (2009) developed a multiscale adjoint algorithm (MSADJ) to compute sensitivity coefficients for high-resolution cases. Just like the MSFV method, the MSADJ method handles the large-scale cases by operating on the same primal coarse volumes for the global coarse-scale adjoint simulation and an auxiliary dual grid for the local fine-scale adjoint simulations. The global adjoint problem is very efficiently solved at the coarse scale and the resulting coarse-scale sensitivities are physically interpolated to the local fine nodes through the retrieval of local variability of model parameters. Therefore, the large-scale problem is divided into a suite of small-scale problems which can be effectively solved at a lower cost by the computational techniques currently available. Moreover, only the coarse-scale forward simulation results are needed to store since the fine-scale fields can be accurately reconstructed with the aid of the interpolators that are defined by local details. The interpolators are selectively stored in memory for later use or can be adaptively, real-time updated when necessary.

Since the fine-scale sensitivity coefficient computed by the multiscale method has a very large dimension and an ordinary gradient-based optimizer cannot accommodate it, a dimension reduction technique is needed for high-resolution inverse modeling. Besides, there exist some artifacts that are introduced by
the multiscale inverse modeling using conventional, gradient-based optimizers (Fu et al., 2009). We reason that these artifacts can be removed if the models’ structure is maintained. In addition, the preservation of the spatial structure helps enhance the reliability of reservoir performance prediction (Fu, 2008). There are several ways to efficiently reduce the dimension of the problem and to maintain the consistency of model structure at the same time, e.g., the pilot point (RamaRao et al., 1995) or the master point method (Gómez-Hernández et al., 1997), the gradual deformation method (Hu, 2000), the probability perturbation method (Caers and Hoffman, 2006), and the (K)PCA-based model reduction method (Sarma et al., 2008), etc. In this work, the gradient-based gradual deformation method is included into the multiscale framework for model updating.

The main contribution of this work is to develop a multiscale framework that seamlessly integrates the multiscale forward simulation to calculate the objective function, the multiscale adjoint model to compute the fine-scale gradient, and the gradient-based gradual deformation method to calibrate the initial models guessed. The framework proposed to attack the large-scale inverse modeling problem is not only accurate and computationally efficient due to the introduction of the multiscale strategy, but also is geologically consistent with the given information thanks to the merit of the gradual deformation method.

The remainder of this paper is organized as follows. In the next section, the multiscale finite-volume method is outlined and applied at solving the single-phase transient flow equation. The multiscale adjoint model is summarized in Section 3. Section 4 presents the details of the gradient-based gradual deformation method for the minimization of the objective function. A synthetic single-phase transient flow example is presented in Section 5 to illustrate the efficiency and powerfulness of the proposed method. Finally, we conclude the paper in Section 6.

2 Multiscale finite-volume method

The MSFV method is applied to solve the single-phase transient flow problem in porous media whose governing equation is given by,

\[ \nabla \cdot (k \cdot \nabla p) = q + s_s \frac{\partial p}{\partial t}, \]

where \( k(x) \) indicates the permeability, \( p(x, t) \) is the pressure, \( q(x, t) \) represents the source and sink term (e.g., wells’ stresses; positive when extracted), and \( s_s \) is the storage coefficient. The objective of the forward simulation is to derive a fine-scale pressure field \( p(x, t) \) from the given fine-scale permeability field \( k(x) \), \( x = (x, y, z) \in \Omega \subset \mathbb{R}^n \). During the computational procedure, a coarse grid \( \bar{\Omega} \subset \mathbb{R}^\bar{n} \) with a block denoted by \( \bar{\omega} \in \bar{\Omega} \) is used to relax the computational burden of the large-scale problem and a dual coarse grid \( \tilde{\Omega} \subset \tilde{\mathbb{R}}^\tilde{n} \) with a block denoted by \( \tilde{\omega} \in \tilde{\Omega} \) is involved to build the physical interpolators and the coarse-scale operator. By connecting the centers of the primal coarse volumes, each
dual coarse volume \( \tilde{\omega} \) consists of \( 2^d \) blocks with a size equal to \( 1/2^d \) of adjacent coarse blocks, \( \tilde{\omega}_i, i \in [1, 2^d] \), where \( d \) is the dimension of the problem. Figure 1 presents a 2D illustration of the primal and the dual coarse grids.

Starting from the fine-scale diagonal tensor model parameters \( k \) at \( \Omega \), a set of interpolators (basis functions) are first built for the fine-scale pressure reconstruction and the equivalent transmissibility tensor \( \bar{T}(x, t), x \in \partial\tilde{\omega} \), is computed for the coarse-scale numerical simulation. The first step is to construct \( 2^d \) sets of dual basis (i.e., weight coefficient) functions, \( \tilde{\varphi}(x, t) = (\tilde{\varphi}^i), i \in [1, 2^d] \), \( x \in \tilde{\Omega} \), one for each adjacent coarse block. Each dual basis function \( \tilde{\varphi}^i \) at \( \tilde{\omega} \) is the solution of the (single-phase-like steady-flow) elliptic problem with respect to \( p \),

\[
\nabla \cdot (k \cdot \nabla p) = 0, \quad x \in \tilde{\omega}.
\]

(2)

For all dual basis functions, \( \tilde{\varphi}^i, i \in [1, 2^d] \), the pressure is set to zero at all the other corners except for the corner \( i \) where \( p_i = 1 \). To ensure the solvability of (2), a boundary condition must be specified ahead. The pressures at boundaries are set as,

\[
\frac{\partial}{\partial x_t} \left( k \cdot \frac{\partial p}{\partial x_t} \right) = 0, \quad x \in \partial\tilde{\omega},
\]

(3)

where \( x_t \) is the coordinate parallel to the boundary \( \partial\tilde{\omega} \).

Then, the equivalent transmissibility tensor \( \bar{T} \) at \( \partial\tilde{\omega} \), which will be used to compute the coarse-scale pressure in the next step, can be assembled from the integral fluxes across the coarse cell interfaces by Darcy’s law,

\[
\bar{T} = -\frac{1}{2} \sum_{i=1}^{2^d} \int_{\partial\omega \cap \tilde{\omega}} k \cdot \nabla \tilde{\varphi}^i d\Gamma, \quad x \in \partial\omega \cap \tilde{\omega},
\]

(4)

Note that the area integration is applied to the entire coarse volume interface in the end. For an incomplete dual volume that locates at the boundary, additional blocks with an extra permeability value are needed to compensate the computation of the dual grid.

Once the equivalent transmissibilities at the coarse scale have been computed by applying (2)-(4) to all dual coarse volumes, the global flow problem can be solved to form the coarse-scale pressure field \( \bar{p}(x, t), x \in \tilde{\Omega} \). The coarse-scale flow problem can be solved by any multi-point stencil finite-volume code using the equivalent transmissibilities for the flux computation, i.e.,

\[
\nabla \cdot (\bar{k} \cdot \nabla \bar{p}) = q + s_s \frac{\partial \bar{p}}{\partial t}, \quad x \in \tilde{\Omega}.
\]

(5)

\( \bar{k} \) is related to \( \bar{T} \) by,

\[
\bar{T} = \bar{k} \frac{\bar{A}_{1/2}}{\Delta x_{1/2}}, \quad x \in \partial\tilde{\omega},
\]
where $\bar{A}_{1/2}$ is the cross-sectional area of two adjacent blocks and $\Delta \bar{x}_{1/2}$ is the distance between them.

The fine-scale pressure field within the dual volume is constructed by,

$$\tilde{p} = \sum_{i=1}^{2^d} \bar{p}_i \bar{\phi}_i, \quad x \in \bar{\omega},$$  \hspace{1cm} (6)

where the pressure $\bar{p}$ is the coarse-scale pressure values from (5).

### 3 Multiscale adjoint-based sensitivity coefficient

Given any initial model, the mismatch between the observations and the simulations computed from (2)-(6) can be used to define an objective function,

$$J = \frac{1}{2}(\zeta(x) - p)^T C_p^{-1}(\zeta(x) - p),$$  \hspace{1cm} (7)

where $\zeta(x)$ is the multiscale forward flow simulator, $x$ is the vector of model parameters (e.g., permeability, porosity, etc.), $p$ represents the observed dynamic (pressure) data, and $C_p$ is the diagonal matrix that reflects the observation errors of dynamic data, $\sigma^2_p$. The objective of this part is to compute the sensitivity coefficients of $J$ to the model parameters $x$ for the gradient-based optimizer. In this work, we assume that $x = \ln k$ is an isotropic field.

#### 3.1 Coarse grid adjoint model

In the multiscale method, the pressure field is actually built at the coarse scale,

$$\tilde{\zeta}_t = \bar{A}_t \tilde{p}_t - \bar{q}_t = \mathbf{0},$$  \hspace{1cm} (8)

where $[\bar{A}_t]_{n \times n}$ is a large sparse matrix pertinent to the transmissibility tensor $T$, $(\tilde{p}_t)_{n \times 1}$ is the unknown pressure vector to be solved, $(\bar{q}_t)_{n \times 1}$ is a known vector that includes boundary conditions, accumulation part, and well-stress, and $(\tilde{\zeta}_t)_{n \times 1}$ is an $n$-dimensional vector at time $t$.

By adjoining the coarse-scale flow equation (8) to the objective function (7), the Lagrangian may be defined as,

$$\tilde{\zeta} = J + \sum_{t=1}^{n_t} \eta_t^T \tilde{\zeta}_t,$$  \hspace{1cm} (9)

where $(\tilde{\eta}_t)_{n \times 1}$ is a vector of Lagrange multipliers. The derivative of the Lagrangian $\tilde{\zeta}$ with respect to the model parameters $(x)_{n \times 1}$ is given by,

$$\frac{d\tilde{\zeta}}{dx} = \frac{dJ}{dx} = \frac{\partial J}{\partial x} + \sum_{t=1}^{n_t} \left( \frac{\partial J}{\partial \tilde{p}_t} \frac{\partial \tilde{p}_t}{\partial x} + \eta_t^T \left( \frac{\partial \tilde{\zeta}_t}{\partial x} + \frac{\partial \tilde{\zeta}_t}{\partial \tilde{p}_t} \frac{\partial \tilde{p}_t}{\partial x} \right) \right),$$  \hspace{1cm} (10)
where the first “=” holds because $\bar{\zeta}_t = \bar{0}$ in (9). Equation (10) can be rearranged as,

$$
\frac{dJ}{dx} = \frac{\partial J}{\partial x} + \sum_{t=1}^{n_t} \left( \frac{\partial J}{\partial p_t} + \bar{\eta}_t^T \frac{\partial \zeta_t}{\partial x} \right) \frac{\partial p_t}{\partial x} + \bar{\eta}_t^T \frac{\partial \bar{\zeta}_t}{\partial x}.
$$

(11)

By forcing,

$$
\sum_{t=1}^{n_t} \left( \frac{\partial J}{\partial p_t} + \bar{\eta}_t^T \frac{\partial \zeta_t}{\partial x} \right) = \bar{0}^T,
$$

(12)

the solutions with respect to $\bar{\eta}_t$ are called the adjoint states and these equations are called adjoint state equations. Note that $\bar{0}^T$ is an $n$-dimensional vector, i.e., $\bar{0} = (0, 0, ..., 0)^T$. The gradient of the objective function $J$ with respect to $x$ is computed by,

$$
\frac{dJ}{dx} = \frac{\partial J}{\partial x} + \sum_{t=1}^{n_t} \bar{\eta}_t^T \frac{\partial \zeta_t}{\partial x},
$$

(13)

Once we have $\bar{\eta}$ by solving (12), $dJ/dx$ can be derived from (13). For each time step $t$ when the adjoint state equation is solved, the vector of Lagrange multipliers $\bar{\eta}_t$ is estimated. Equation (12) at any time $t$ can be written as,

$$
\begin{bmatrix}
\frac{\partial \zeta_t}{\partial x} \\
\frac{\partial \zeta_t}{\partial p_t}
\end{bmatrix}^T \bar{\eta}_t = - \left( \frac{\partial J_t}{\partial p_t} \right)^T,
$$

(14)

subject to the following initial and boundary conditions,

$$
\bar{\eta}(x, n_t + 1) = \bar{0}, \quad x \in \bar{\Omega},
$$

$$
\bar{\eta}(x, t) = 0, \quad x \in \partial \bar{\Omega}_D,
$$

$$
\frac{\partial \bar{\eta}(x, t)}{\partial n} = 0, \quad x \in \partial \bar{\Omega}_N.
$$

One can easily find that (14) has a very similar form as the flow equation (8) but backwards in time, so it can be solved using the same subroutine with a slight change. Note that the initial adjoint state $\bar{\eta}_{n_t+1} = \bar{0}$.

### 3.2 Subgrid adjoint model

One term in (13), $\partial \zeta_t / \partial x$, is needed to expand according to the flow equation (8),

$$
\begin{bmatrix}
\frac{\partial \zeta_t}{\partial x} \\
\frac{\partial \zeta_t}{\partial T_t} \\
\frac{\partial \zeta_t}{\partial p_t}
\end{bmatrix}^T = \begin{bmatrix}
\frac{\partial T_t}{\partial x} \\
\frac{\partial T_t}{\partial T_t} \\
\frac{\partial T_t}{\partial p_t}
\end{bmatrix}^T = \begin{bmatrix}
\frac{\partial T}{\partial x} \\
\frac{\partial T}{\partial T} \\
\frac{\partial T}{\partial p}
\end{bmatrix}^T
$$

(15)

where we drop the subscript $t$ for the last “=” just for the convenience of presentation. The computation of the first matrix $[\partial T / \partial x]$ uses the dual basis
functions and the second matrix $[\partial \bar{\zeta} / \partial \bar{T}]$ can be expanded directly according to (8) without any difficulty. Note that there is a scale disparity in $\partial \bar{T} / \partial x$ which comes from the “upscaling” procedure. Moreover, $\partial \bar{T} / \partial x$ is independent on $t$ for the single-phase cases. According to (4), the equivalent transmissibility is,

$$
\bar{T} = -\frac{1}{2} \sum_{i=1}^{2^d} \int_{\partial \bar{\omega} \cap \tilde{\omega}} k \cdot \nabla \bar{\varphi}_i d\Gamma, \quad \mathbf{x} \in \partial \bar{\omega} \cap \tilde{\omega},
$$

subject to,

$$
\nabla \cdot (k \cdot \nabla \bar{\varphi}_i) = 0, \quad \mathbf{x} \in \tilde{\omega},
$$

with the reduced boundary conditions specified by,

$$
\frac{\partial}{\partial x_t} \left( k \cdot \frac{\partial \bar{\varphi}_i}{\partial x_t} \right) = 0, \quad \mathbf{x} \in \partial \tilde{\omega}.
$$

Although there are several alternatives to compute $[\partial \bar{T} / \partial x]$, one of the efficient ways is to use the adjoint method the second time, which is embedded in the primal coarse-scale adjoint-sensitivity formulation (12)-(13). The discretized equation of (17) for any dual basis function $\tilde{\varphi}_i$ is,

$$
\tilde{\varsigma} = \tilde{A} \tilde{\varphi} - \tilde{q} = \tilde{0},
$$

where $\tilde{A} = \tilde{A}(\mathbf{x})$ is the conductance matrix, $\tilde{\varphi}$ is the vector of constructed dual basis functions, and $\tilde{q}$ is the vector of source/sink terms which obviously come from the specified pressure pulses in computing the basis functions. Since our objective is to compute $[\partial \bar{T} / \partial x] = [\partial \bar{T}(\varphi) / \partial x]$, we can build a Lagrangian for any and every component of $\bar{T}$,

$$
\ell = \bar{T} + \tilde{\eta}^T \bar{\zeta},
$$

where $\tilde{\eta}$ is the adjoint state nodal vector. Taking its derivative with respect to the model parameters $x$ leads to,

$$
\frac{d\ell}{dx} = \frac{d\bar{T}}{dx} = \frac{\partial \bar{T}}{\partial x} + \frac{\partial \bar{T}}{\partial \bar{\varphi}} \frac{\partial \bar{\varphi}}{\partial x} + \tilde{\eta}^T \left( \frac{\partial \bar{\zeta}}{\partial x} + \frac{\partial \bar{\zeta}}{\partial \bar{\varphi}} \frac{\partial \bar{\varphi}}{\partial x} \right).
$$

The adjoint state $\tilde{\eta}$ is chosen such that,

$$
\frac{\partial \bar{T}}{\partial \bar{\varphi}} + \tilde{\eta}^T \frac{\partial \bar{\zeta}}{\partial \bar{\varphi}} = \tilde{0}^T, \quad (19)
$$

where $\bar{T}$ is related to $\bar{\varphi}$ through (16). Obviously, an equivalent steady-state flow problem is needed to solve in order to obtain $\tilde{\eta}$. The boundary condition for (19) is simply set as $\tilde{\eta}(\mathbf{x}) = 0, \mathbf{x} \in \partial \bar{\omega}$, for the Dirichlet boundary and, at the same time, (13) is required to satisfy in building the adjoint flow equation. Therefore, $[\partial \bar{T} / \partial x]$ can then be computed by the marginal sensitivity,

$$
\frac{dT}{dx} = \frac{\partial \bar{T}}{\partial x} + \tilde{\eta}^T \frac{\partial \bar{\zeta}}{\partial x}.
$$

(20)
The major computational cost of this part is $2^{d-1}d$ equivalent forward simulations for each dual basis function with most of resources consuming by (19). Each equivalent forward simulation corresponds to one part of equivalent coarse-scale transmissibilities. Note that, in one dual block, there are $d$-dimension-dependent $\bar{T}$ components, (say, $\bar{T}_{xx}$, $\bar{T}_{yy}$, and $\bar{T}_{zz}$ for a 3D case), with each component having $2^{d-1}$ sub-faces or segments that belong to distinct primal coarse-scale $\bar{T}$. However, one can easily find that (19) is actually independent on the basis functions, which means that only $2^{d-1}d$ equivalent forward simulations for one dual block are required.

4 Reparameterization by gradient-based gradual deformation

In order to minimize the objective function $J$ in (7), the gradual deformation method continuously updates a multiGaussian field $x = \ln k$ according to,

$$x_l = x_{l-1} \cos \alpha + u_l \sin \alpha,$$

(21)

where $x_l$ is the updated field at the iteration $l$, $x_{l-1}$ is the old field at the previous iteration, $u_l$ is an auxiliary field that is used to update $x_{l-1}$ for the iteration $l$, and $\alpha$ is a new parameter that controls the updating. Obviously, $\alpha = 0$ means that the updating is zero and the field remains unchanged, i.e., $x_l = x_{l-1}$; $\alpha = \pi/2$ means an entirely new updating, i.e., $x_l = u_l$. The choice of coefficients $\cos \alpha$ and $\sin \alpha$ for the linear combination (21) ensures the preservation of the spatial variability (e.g., the variogram) during the updating.

The problem now turns to find an optimal $\alpha_l^{opt}$ such that the objective function is minimized, i.e., $\alpha_l^{opt} = \text{argmin} J$, for the given auxiliary field $u_l$. This essentially is a reparameterization procedure since the original large-scale multi-dimensional problem (i.e., to find $\Delta x_l^{opt}$) is transformed to the current one-dimensional problem (i.e., to find $\alpha_l^{opt}$) for optimization. The derivative of the objective function $J$ with respect to $\alpha$ is undoubtably helpful for the one-dimensional optimization problem. The gradient of the objective function is,

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial x_l} \frac{\partial x_l}{\partial \alpha},$$

(22)

where $(\partial J/\partial x_l)$ is the sensitivity coefficient and $(\partial x_l/\partial \alpha)$ is computed from (21) by,

$$\frac{\partial x_l}{\partial \alpha} = -x_{l-1} \sin \alpha + u_l \cos \alpha.$$

(23)

On the basis of $x_{l-1}$ and $u_l$, the optimal $\alpha$ for the iteration $l$ can be found as follows. Starting from any $\alpha = \alpha_0$, say $\alpha_0 = 0$, one can obtain an initial field $x_l|_{\alpha=\alpha_0}$ from (21). Running the forward and adjoint simulators will yield the derivative of the objective function with respect to $x_l$, i.e., $(\partial J/\partial x_l)|_{\alpha=\alpha_0}$.
From (23) and (22), one can easily obtain \((\partial x_l/\partial \alpha)|_{\alpha=\alpha_0}\) and \((dJ/d\alpha)|_{\alpha=\alpha_0}\), respectively. With (22), any gradient-based search algorithm (e.g., the steepest descent method, the conjugate gradient method, the Newton method, etc.) can be used to locate the optimal estimate \(\alpha\) by repeating the procedure as above until convergence.

In (21), \(u_l\) is stochastically generated and totally independent, which makes model updating too random and thus convergence slow. A more efficient alternative is to select \(u_l\) as close as possible to the gradient search direction. Hu and Le Ravalec-Dupin (2004) proposed to use a group of independent realizations \(v_i, i \in [1, n_g]\), to replace \(u_l, u_l = \sum_{i=1}^{n_g} \lambda_i v_i, \sum_{i=1}^{n_g} \lambda_i^2 = 1,\)

where the coefficient \(\lambda_i\) can be computed by,

\[ \lambda_i = \frac{\langle g, v_i \rangle}{\sum_{j=1}^{n_g} \langle g, v_j \rangle^2}, \quad i \in [1, n_g]. \]

where \(\langle \cdot, \cdot \rangle\) denotes the inner product and \(g^T = (\partial J/\partial x_l)\) is the marginal sensitivity coefficient from (13). Note that the construction of \(u_l\) does not need any run of the flow simulation and, thus, is very cheap.

A complete procedure for the multiscale inverse modeling can be summarized as follows,

- **Step 1**: \(x_l \leftarrow x_0\)
- **Step 2**: Compute \(J(x_l)\)
- **Step 3**: Loop outer (find \(x_{opt}\)): If \(J(x_l) > \varepsilon_J\),
  - Step o.1: Compute \(g = J'(x_l)\)
  - Step o.2: \(x_{l-1} \leftarrow x_l\)
  - Step o.3: Generate \(v_i, i \leq n_g\)
  - Step o.4: Compute \(\lambda_i = \langle g, v_i \rangle / \sum_{j=1}^{n_g} \langle g, v_j \rangle^2, i \leq n_g\)
  - Step o.5: Compute \(u_l = \sum_{i=1}^{n_g} \lambda_i v_i\)
  - Step o.6: \(\alpha_{m-1} = 0\)
- **Loop inner (find \(\alpha_{opt}\))**: If \(\Delta \alpha_m > \varepsilon_\alpha\),
  * Step i.1: \(\alpha_m = \alpha_{m-1} + \Delta \alpha_m\)
  * Step i.2: \(x_l = x_{l-1} \cos \alpha_m + u_l \sin \alpha_m\)
Step i.3: \( J(\alpha_m) = J(x_l) \)
Step i.4: \( J'(\alpha_m) = J'(x_l)(-x_{l-1} \sin \alpha_m + u_l \cos \alpha_m) \)
Step i.5: Compute \( \Delta \alpha_m \)
Step i.6: \( \alpha_{m-1} \leftarrow \alpha_m \)
- Loop inner (end find \( \alpha_{opt} \))
  - Loop outer (end find \( x_{opt} \))
  - Step 4: Output \( x_l \)

5 An illustrative example

5.1 Experimental configuration

A synthetic example problem is presented to show the power of the multiscale inverse modeling method proposed. The illustrative example has \( 144 \times 81 \times 1 \) fine cells (of order \( 10^5 \)). The size of each cell is \( 10.0 \times 10.0 \times 20.0 \). The fine grid is upscaled to \( 16 \times 9 \times 1 \) coarse blocks with each one having \( 9 \times 9 \times 1 \) fine cells for the multiscale simulation. There are two horizontal wells that control the fluid flow "uniformly" from left to right: the left side has a constant pressure 15.0, the right side has a constant pressure 0.0. The initial pressure distribution is zero everywhere except for the left side where the pressure remains unchanged (15.0) during the entire simulation period. The time discretization uses the so-called time multiplier scheme,

\[
\Delta t_i = \alpha \Delta t_{i-1},
\]

which allows a smaller time-step at the early stage in order to reduce the modeling error. The entire time span is divided into 100 time steps for numerical simulation with the time-step multiplier \( \alpha = 1.15 \) in this case study. The porosity is constant and the storage coefficient is \( 10^{-5} \). Therefore, this case simulates the transient model response (pressure) that is intrigued by a fluid injection-extraction operated on the confined reservoir.

The transient pressure data are continuously collected at four wells (see Figure 2(A) for the “true” \( \ln k \) field and the wells’ positions) which serve as conditioning data for inverse conditional modeling (see Figure 4 for corresponding curves denoted by “observation”). Note that the conditioning data are extracted from the “true” reference field. The objective of inverse conditional modeling, therefore, is to tune any given initial \( \ln k \) field (e.g., Figure 2(B)) to match the four observed pressure history curves using the multiscale method developed. Both the reference and the initial \( \ln k \) field (Figure 2) are generated by a public domain program GCOSIM3D (Gómez-Hernández and Journel, 1993). An exponential variogram is specified in generating the two isotropic fields. All the initial and the boundary conditions are assumed to be identified unbiasedly; in other words, the unique uncertain source is the \( \ln k \) field.
5.2 Inverse modeling

Figure 3 shows the evolution history of the objective function (dashed line) using the proposed multiscale inverse modeling method. As a reference, the objective function evolution by the conventional single-scale (i.e., fine-scale) adjoint method (solid line) is also plotted. Obviously, they share a similar convergence performance. A well matched model has been obtained in three to four iterations using both methods. To further validate the accuracy of the multiscale method developed, a set of additional fine-scale simulations are applied to the multiscale updated models. The evolution history of the extra fine-scale objective functions is plotted in Figure 3 by the dotted line. One can easily find that most of multiscale results reasonably approximate the fine-scale pressure responses. Moreover, it seems that the absolute error decreases as the iteration proceeds. Note that the fluctuation of the inconsistency between the MSADJ and the MSADJ-FSim gradually flats down as the mismatch decreases.

Since there is no much significant improvement after four iterations, we take the model at the iteration #4 as the final, matched model. Figure 4 compares the pressure history curves of the multiscale matched model at all the four observation well points. One can find that the multiscale simulations are very close to the fine-scale simulations and the given observation data. Compared to the initial model, moreover, the observations are well reproduced by the multiscale matched model. A detailed error analysis on the final, multiscale matched model (i.e., iteration #4) is listed on Table 1. Compared to the initial misfit, the absolute error of the individual wells’ mismatch between the multiscale (denoted by MsGGD) and the fine-scale (denoted by MsGGD-Fsim) simulations is insignificant. The total error of the objective function is only 2.699, less than 1% compared to the initial mismatch (290.288), which displays a sufficient accuracy. The result is quite close to that of the fine-scale inverse modeling using the same iterations (denoted by FsGGD).

Moreover, the models generated as such are geologically consistent, which means that the model’s spatial structure is preserved during the inverse modeling procedure if the structure parameters are “correctly” identified. For a multi-Gaussian field, the main structure parameters are represented by the first two moments, i.e., mean, variance (or standard deviation) and covariance. Figure 6 compares the histogram reproduction of the conditional and non-conditional models. One can easily find that the matched models reasonably reproduce the a priori histogram, especially the standard deviation. On the other hand, both the multiscale and the fine-scale methods do display a certain self-adjust ability to fit to the reference field since the mean and variance of the initial field (i.e., a priori) is deviated from those of the reference. This point can be seen more clearly from the reproduction of the mean. Figure 7 compares the variogram reproduction of the conditional and non-conditional models. Just like the fine-scale method, the multiscale method yields a quite identical distribution since the initial and the reference fields well match to each other. Of course,
it is worth pointing out that the reproduction of the prior information is also dependent on the experimental network design. In summary, the multiscale inverse modeling proposed efficiently improves the reproduction of the observation data at all the well points. The inconsistency between the multiscale and the real, fine-scale simulations is minor and thus negligible for inverse modeling problems. Moreover, the models generated are geologically consistent.

5.3 GGD versus BFGS

It is interesting to compare the gradient-based gradual deformation (GGD) method to other traditional optimization algorithms, e.g., the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method in performing the multiscale inverse modeling. The BFGS method (also called quasi-Newton method) is one of the traditional gradient-based optimization methods. Its key feature is to iteratively build up a good approximation to the inverse Hessian matrix (Press et al., 1992).

Within this method, the model is updated by,

\[ x_l = x_{l-1} + H_l(g_l - g_{l-1}), \]  

(24)

where the Hessian matrix is computed by,

\[
H_l = H_{l-1} + \frac{(x_l - x_{l-1}) \otimes (x_l - x_{l-1})}{(x_l - x_{l-1})^T (g_l - g_{l-1})} - \frac{[H_{l-1}(g_l - g_{l-1})] \otimes [H_{l-1}(g_l - g_{l-1})]}{(g_l - g_{l-1})^T H_{l-1}(g_l - g_{l-1})}
\]

\[ + (g_l - g_{l-1})^T H_{l-1}(g_l - g_{l-1}) u \otimes u, \]

where \( \otimes \) denotes the outer product, \( g_l^T = (\partial J/\partial x_l) \) is the marginal sensitivity coefficient, and \( u \) is defined by,

\[
u = \frac{(x_l - x_{l-1})}{(x_l - x_{l-1})^T (g_l - g_{l-1})} - \frac{H_{l-1}(g_l - g_{l-1})}{(g_l - g_{l-1})^T H_{l-1}(g_l - g_{l-1})}.\]

One can easily find that, during the model calibration, the prior information never enters the model updating. Note that neither \( H_l \) nor \( g_l \) in (24) contains the prior information. Therefore, this method fails to preserve the prior spatial structure.

Figure 8 compares the evolution history of the objective function obtained by the BFGS-based multiscale method. The convergence velocity is slightly slower than the GGD-based method (see Figure 3 for a comparison). Four to five iterations are needed to achieve a reasonably matched model. The reproduction of wells’ observations is plotted in Figure 9. Compared to the initial model, the matched models better reproduce the observation data. The inconsistency between the multiscale and the fine-scale modeling is minor but worse than the GGD-based method (see Figure 4 for a comparison). This is partially because the model updating by the BFGS-based method only focuses on tuning the local details around the observation well points which are often the most sensitive
locations. The singularity of heterogeneity obviously worsens after parameter tuning, while the forward and adjoint simulations indeed suffer more from the higher heterogeneity. In other words, the model generated by the BFGS-based method is not geologically consistent. This point can be seen more clearly by observing the ln$k$ field generated.

Figure 10 plots the ln$k$ field matched by the BFGS-based multiscale and fine-scale methods. One can easily find an artifact caused by the multiscale method in Figure 10(A). The trace of “coarse-block” can be clearly seen around the well #1. Although accurate enough, the multiscale method is indeed an approximate proxy. Fortunately, this artifact can be suppressed by a global perturbation-based, structure-preservation inverse modeling algorithm, e.g., the gradual deformation method.

Figure 11 shows the model updating after matching to the pressure data, i.e., the perturbations of model parameters compared to the initial guessed model, which is obtained simply by subtracting the initial model from the matched model. The GGD-based perturbations (Figures 11(A) and 11(B)) evidently demonstrate a global pattern, while the BFGS-based perturbations are local and thus is not geologically consistent (Figures 11(C) and 11(D)). More interesting, in the inverse-conditional model, there is no any artifact introduced by the multiscale approximation, nor any singularity of heterogeneity, actually. Indeed, compared to Figure 11(D), Figure 11(C) shows obvious artifacts around wells; but comparing to Figure 11(B), one can not find any “coarse-block” artifact in Figure 11(A). These artifacts are successfully “smeared out” by the GGD-based method simply because of the preservation of the prior information.

In summary, the spatial structure preservation helps suppress the artifacts caused by the multiscale inverse modeling. The GGD-based multiscale inverse modeling is preferred because it preserves the prior information and also because it reduces the dimension of the optimization problem to one dimension. The BFGS-based method obviously fails to do so.

5.4 Model response prediction

Although there is still a debate on whether or not preserving model structure for inverse modeling (e.g., the so-called geologically consistent history matching), Fu (2008) argued that the preservation of “correct” prior information for inverse stochastic modeling actually helps enhance the prediction ability of the models generated, especially for predicting new reservoir performances at new wells. In this part, we will examine the prediction ability of the multiscale matched model.

We take the water cut (i.e., the breakthrough curve (BTC)) as a metric to measure the reliability of models rendered by inverse conditional modeling constraint to pressure data. This is because the water cut as a transport response is more sensitive to the spatial variation of model than the pressure as a flow response. Specifically, a conservative tracer test is designed and imposed on the assumed reservoir. After the reservoir operation arrives at the steady state under the same boundary conditions as before, four thousand passive particles
are released at the left boundary (i.e., the left horizontal controlling well) and are tracked until they arrive at the right boundary (i.e., the right horizontal controlling well). The release source of particles covers the whole left boundary (i.e., a line source) and the particles almost distribute over the entire domain of the field when they move along with the fluid flow such that the particles can largely sample the entire \( l \) field and the overall spatial variability can be captured. All the computations are carried out on the fine scale just for comparing the real prediction ability of the various models generated. A random walk particle tracking method is used to solve the transport equation (Fu, 2008).

The breakthrough curve observed at the right side of the reservoir is recorded and plotted in Figure 12. The curves from the reference, initial, fine-scale and BFGS-based methods are also plotted as a comparison. One can easily find that the GGD-based multiscale methods obviously outperform the BFGS-based methods in predicting the water cut. The BFGS-based methods only slightly enhance the prediction ability on BTC compared to the initial field. This is reasonable because the BFGS-based methods only focus on tuning the local details around the conditioning points; see Figures 11(C) and 11(D) for their differences from the initial field. Moreover, the difference between the multiscale matched and the fine-scale matched models is relatively small for both the GGD-based and the BFGS-based methods, although it is quite unfair to completely compare them because they have different mismatches. Note that the better matches (i.e., the smaller objective functions) are obtained by the fine-scale modelings than the multiscale modelings, after all; see Table 1 for the detailed mismatch comparison.

We should admit that this finding is subject to a further verification through the quantification of the uncertainty reduction owing to the inverse-conditional modeling, e.g., performing a Monte Carlo simulation by running an ensemble of inverse conditional realizations. This evidently entails an inverse stochastic modeling method that is capable of generating multiple inverse conditional realizations (Fu, 2008), rather than only one realization by the method presented. Anyhow, the multiscale method presented is an efficient means that leads us to the plausible model space, at least. Further details are outside of the present paper.

6 Conclusions

A multiscale framework is presented to address the large-scale inverse modeling problem. The inverse modeling problem is cast as an optimization problem and the gradient information is used to efficiently update the model for minimizing the objective function. The large-scale problem is addressed using the multiscale method which attains a sufficiently accurate approximation to the fully fine-scale simulation. Within this framework, the model responses are very efficiently and accurately computed from the fine-scale model using the multiscale finite-volume method. The mismatch between the observations and the simulations is then defined as the objective function and the fine-scale sensitivity
coefficient is computed by the multiscale adjoint method for the subsequent optimization. The difficult, high-dimensional optimization problem is reduced to a one-dimensional one using the gradient-based gradual deformation method.

A synthetic example demonstrates the power of the proposed method. The inverse modeling is very economical and efficient since the fully fine-scale forward and adjoint simulations never intervene in this procedure. Rather, all the forward and adjoint simulations are operated in a multiscale manner. This is the main reason why this multiscale framework can successfully solve the large-scale inverse modeling problem. The inverse-conditional model generated by this method sufficiently attains the fine-scale approximation. This point has been clarified by comparing the multiscale results to the fine-scale ones in the synthetic example. Moreover, this method is natural to the massively parallel computation, which is very useful for assembling individual computational resources to enhance the computational velocity. Yet, this point has not been validated in this work.

The gradient-based gradual deformation algorithm plays an important role in this multiscale framework. It not only reduces the difficult, large-scale optimization problem to a one-dimensional one, but also preserves the prior information for inverse modeling. The preservation of prior information is very important for inverse modeling since it might be the only way (so far to us) to generate geologically consistent inverse-conditional realizations. Actually, geologically consistent inverse-conditional modeling helps improve the prediction ability of models if the prior information well represents the subsurface reality. An additional tracer test demonstrates that this point is very important for the accurate prediction of the transport features of reservoir.

Another interesting finding is that the prior information preservation successfully suppresses the artifacts introduced by the multiscale inverse modeling. Using the traditional gradient-based optimization methods, the model after matching to the state data displays a significant “coarse-block” artifact in the region around the conditioning data, e.g., the well points. This is because the multiscale method in essence operates the global forward and adjoint simulations at the coarse-scale blocks and then interpolates the coarse-scale information into the fine-scale cells. The error caused by the multiscale method is unfortunately magnified by the inverse modeling because the parameter tuning is very sensitive to this approximation. Indeed, the sensitivity coefficients are much smaller than the model parameters, often in several orders. A relatively small parameter perturbation in model might, in pattern, give rise to a significant change in sensitivity coefficient. The multiscale method presented in this work completely circumvents this problem by maintaining the consistence of the model structure.

Acknowledgements

A postdoctoral financial support to the first author from SCRF and SUPRI-B is gratefully acknowledged. We thank Dr. Lin Hu for a useful discussion on the implementation of the gradual deformation method.
References


Table 1: *Mismatch and objective function*

<table>
<thead>
<tr>
<th>Scenario</th>
<th>well #1</th>
<th>well #2</th>
<th>well #3</th>
<th>well #4</th>
<th>total misfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial</td>
<td>254.407</td>
<td>23.112</td>
<td>6.074</td>
<td>6.635</td>
<td>290.228</td>
</tr>
<tr>
<td>MsGGD</td>
<td>0.863</td>
<td>3.870</td>
<td>0.477</td>
<td>5.269</td>
<td>10.479</td>
</tr>
<tr>
<td>MsGGD-Fsim</td>
<td>3.035</td>
<td>3.986</td>
<td>0.803</td>
<td>5.353</td>
<td>13.178</td>
</tr>
<tr>
<td>FsGGD</td>
<td>0.723</td>
<td>4.675</td>
<td>1.555</td>
<td>5.703</td>
<td>12.656</td>
</tr>
<tr>
<td>MsBFGS</td>
<td>7.442</td>
<td>4.322</td>
<td>1.028</td>
<td>7.599</td>
<td>20.392</td>
</tr>
<tr>
<td>MsBFGS-Fsim</td>
<td>0.792</td>
<td>5.507</td>
<td>2.059</td>
<td>6.688</td>
<td>15.046</td>
</tr>
<tr>
<td>FsBFGS</td>
<td>1.107</td>
<td>4.631</td>
<td>1.149</td>
<td>6.845</td>
<td>13.733</td>
</tr>
</tbody>
</table>
Figure 1: Fine grid ($\omega \in \Omega$), coarse grid ($\bar{\omega} \in \bar{\Omega}$), and dual grid ($\tilde{\omega} \in \tilde{\Omega}$)
Figure 2: \( \ln k \) fields: (A) reference and (B) initial
Figure 3: Evolution of the objective functions (GGD-based): FSADJ shows fine-scale adjoint method for reference, MSADJ represents the multiscale adjoint method, and MSADJ-FSim represents the fully fine-scale simulation of the multiscale matched model.
Figure 4: Well pressure history (GGD-based): “observation” means the given observation data, “initial” means the initial realization, “MSADJ” represents the multiscale adjoint method, and “MSADJ-FSim” represents the fully fine-scale simulation of the multiscale matched model.
Figure 5: Matched lnK fields by GGD-based methods: (A) multiscale and (B) fine-scale
Figure 6: lnK histogram distribution: (A) reference, (B) initial, (C) multiscale, and (D) fine-scale
Figure 7: In k variogram distribution: (A) x-direction and (B) y-direction
Figure 8: Evolution of the objective functions (BFGS-based): FSADJ shows fine-scale adjoint method for reference, MSADJ represents the multiscale adjoint method, and MSADJ-FS represents the fully fine-scale simulation of the multiscale matched model.
Figure 9: Well pressure history (BFGS-based): “observation” means the given observation data, “initial” means the initial realization, “MSADJ” represents the multiscale adjoint method, and “MSADJ-FSim” represents the fully fine-scale simulation of the multiscale matched model.
Figure 10: Matched lnK fields by BFGD-based methods: (A) multiscale and (B) fine-scale
Figure 11: $\ln k$ perturbations after matching: (A) multiscale GGD-based method, (B) fine-scale GGD-based method, (C) multiscale BFGS-based method, and (D) fine-scale BFGS-based method. Note that the images are obtained simply by subtracting the initial image from the matched images.
Figure 12: A comparison on breakthrough curves: “reference” means the reference field, “initial” means the initial realization, “MsGGD” denotes the GGD-based multiscale matched realization, “FsGGD” represents the GGD-based fine-scale matched realization, “MsBFGS” denotes the BFGS-based multiscale matched realization, and “FsBFGS” represents the BFGS-based fine-scale matched realization. Note that all the simulations are run at the fine scale.