SPE 119112

Adjoint Multiscale Mixed Finite Elements

S. Krogstad, V.L. Hauge, A.F. Gulbransen, SINTEF ICT

This paper was prepared for presentation at the 2009 SPE Reservoir Simulation Symposium held in The Woodlands, Texas, USA., 2–4 February 2009.

Abstract

We develop an adjoint model for a simulator consisting of a multiscale pressure solver and a saturation solver that works on flow adapted grids. The multiscale method solves the pressure on a coarse grid that is close to uniform in index space and incorporates fine-grid effects through numerically computed basis functions. The transport solver works on a coarse grid adapted by a fine-grid velocity field obtained by the multiscale solver. Both the multiscale solver for pressure and the flow-based coarsening approach for transport have earlier shown the ability to produce accurate results for high degree of coarsening. We present results for a complex realistic model to demonstrate that control settings based on optimization of our multiscale flow-based model closely matches or even outperforms those found by using a fine-grid model. For additional speed-up, we develop mappings used for rapid system updates during the time-stepping procedure. As a result, no fine-grid quantities are required during simulations and all fine grid computations (multiscale basis functions, generation of coarse transport grid and coarse mappings) become a preprocessing step. The combined methodology enables optimization of water flooding on a complex model with 45,000 grid-cells in a few minutes.

Introduction

The ability to perform fast reservoir simulation is crucial for optimization workflows within real-time/closed-loop reservoir management. Current simulators are far from meeting these requirements if detailed geological information is to be utilized. In optimization loops in which the reservoir simulation is just one of multiple function evaluations, the need for model-reduction techniques to reduce the computational load becomes inevitable. Recently, so-called reduced order modeling techniques using proper orthogonal decompositions (POD) have received great interest in reservoir simulation research (see e.g., van Doren et al. (2006), Cardoso et al. (2008), and references therein). The POD-based techniques generate a reduced-order basis for the states based on snapshots from an initial full-order simulation, and have been shown to produce accurate results and give speedup factors up to two orders of magnitude (Cardoso et al. 2008).

Gradient-based optimization of production settings using the adjoint model dates back to works by Ramirez (1987) and Asheim (1987), and have received a lot of interest recent years, starting with Brouwer and Jansen (2004). The adjoint model is a means to efficiently compute gradients of an objective function and has been suggested both for production optimization and history matching (see e.g, Sarma et al. (2006) for an application of the adjoint model to closed-loop reservoir management). In most applications fully-implicit formulations have been used in which the coefficient matrix of the adjoint equations is just the transpose of the Jacobian in the forward system (Li et al. 2003).

Multiscale modeling of flow in porous media has become an active research area in recent years. Common for these methods is that they seek efficient solutions of equations with rough coefficients without scale separation, and as other model-reduction techniques seek solutions in low dimensional spaces. However, the bases in the multiscale approach are constructed by solving a number of local fine-scale flow problems rather than the full fine-scale problem as in POD. The starting point of much of the relevant multiscale modeling research was the paper of Hou and Wu (1997). Among active research topics today is the multiscale finite-volume method, which first appeared in Jenny et al. (2003), and the multiscale mixed finite-element method (MsMFEM) originating from Chen and Hou (2003). In this work we will employ a recent version of the MsMFEM (Aarnes et al. 2008) in conjunction with a coarse saturation solver suggested by Aarnes et al. (2007) as a model-reduction technique for optimization of water flooding. The main new contributions of this paper is the development of an adjoint formulation for this methodology and coarse grid mappings to accelerate the computations involved.

The paper proceeds as follows: we start by introducing the fine-grid discretization of the model problem in some detail before we introduce the multiscale and coarse-grid counterparts. We then develop the adjoint models corresponding to both the fine-grid and coarse-grid models and show that these have basically the same structure as the forward models. In a brief section

we comment on accelerating the computations involved in the coarse forward and adjoint model, and finally we illustrate the suggested methodology in two numerical examples. The first is a validation example considering 2D layers, and the second considers a complex geological model using varying levels of coarsening. All the coarse models show good performance, even the coarsest which requires less than 5 seconds for each forward simulation on a 45,000 grid-cell model.

Model and Fine-Scale Discretization

In this work we consider water-flooding examples, and restrict the model to two-phase incompressible flow neglecting gravity and capillary forces. We note, however, that the multiscale methodology employed here is also extended to handle more physical effects (Krogstad et al. 2009).

Let Ω denote the computational domain with boundary Γ. The *pressure equation* gives the Darcy velocities $\vec{v}$ and pressure p for a given water saturation field s:

$$\vec{v} = -\lambda(s)\mathbf{K}\nabla p, \qquad \nabla \cdot \vec{v} = q \quad \text{in} \quad \Omega, \tag{1}$$

with boundary conditions $\vec{v}\cdot\vec{n} = v_N$ on Γ_N and $p = p_D$ on Γ_D, where $\vec{n}$ is the outward pointing unit normal. In Eq. 1, $\mathbf{K}$ is the permeability tensor, $\lambda(s)$ is the total mobility, and q is the source term. The corresponding *saturation equation* is then given as

$$\phi\frac{\partial s}{\partial t} + \nabla\cdot(f(s)\vec{v}) = q_w, \tag{2}$$

where ϕ is the porosity, f is the water fractional flow function, and q_w is the water source term. In the following, we represent wells as boundary conditions. Hence, a well w with boundary γ_w is conceptually represented as a *hole* in Ω with $\gamma_w \subset \Gamma$. We consider wells that are either pressure constrained or rate constrained, and accordingly either $\gamma_w \subset \Gamma_D$ with $p = p_D$ on γ_w or $\gamma_w \subset \Gamma_N$ with $\int_{\gamma_w}\vec{v}\cdot\vec{n} = -q_w$.

Discretization and Hybrid System. We start by discussing the fine-grid discretization of Eq. 1 in some detail. Let $\{E_i\}$ be a set of N polyhedral cells constituting a grid for Ω. For a given cell E with faces e_k, $k = 1,\ldots,n_E$, let $\boldsymbol{v}_E$ be the vector of outward pointing fluxes of the corresponding faces of E, p_E the pressure at the cell center, and $\boldsymbol{\pi}_E$ the pressures at the cell faces. Most discretization methods used for reservoir simulation relate these quantities through a *transmissibility* matrix $\boldsymbol{T}_E$, such that

$$\boldsymbol{v}_E = \lambda(s_E)\boldsymbol{T}_E(p_E - \boldsymbol{\pi}_E). \tag{3}$$

Examples include the two-point flux-approximation (TPFA) method (see e.g., Aziz and Settari 1979), the lowest-order mixed finite element methods (MFEM) (see e.g., Brezzi and Fortin 1991), and recent mimetic finite-difference methods (MFDM) by Brezzi et al. (2005). While MFEM and MFDM in general lead to full $\boldsymbol{T}_E$-matrices, the TPFA results in a diagonal $\boldsymbol{T}_E$, but is nonconvergent for general grids. We refer the reader to the cited references for details on the computation of $\boldsymbol{T}_E$ for each method. The MFDM efficiently handles degenerate hexahedral cells arising in the corner-point format and non-matching interfaces that occur across faults, and was used as a fine-grid solver by Aarnes et al. (2008).

Wells are modeled using well indices (Peaceman 1983), and thus if a gridcell E is perforated by a well w, the set of local equations in Eq. 3 is extended by the equation

$$-q_E^w = \lambda(s_E)\, WI_E^w(p_E - p_E^w). \tag{4}$$

Here WI_E^w is the well index, p_E^w is the perforation well pressure, and p_E is the numerically computed pressure in the perforated cell E. Note that conceptually, adding Eq. 4 to Eq. 3 can be regarded as adding an extra face to the polyhedra E. Accordingly, wells are considered as boundary faces, and well pressure and rate constraints are implemented as Dirichlet and Neumann boundary conditions, respectively.

The local equations given by Eqs. 4 and 3 are joined together in a global *hybrid* system. Let $\boldsymbol{v}$ denote the outward face fluxes and well perforation rates (with negative signs) ordered cell-wise (thus, fluxes on interior faces appear twice with opposite signs), $\boldsymbol{s}$ the cell saturations, $\boldsymbol{p}$ the cell pressures, and $\boldsymbol{\pi}$ the face pressures and well pressures. Assuming zero-flux boundary conditions except at wells, the system corresponding to the pressure equation at time t^n takes the form

$$\begin{bmatrix} \boldsymbol{B}(\boldsymbol{s}^{n-1}) & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{D}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^n \\ -\boldsymbol{p}^n \\ \boldsymbol{\pi}^n \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{v}_N^n \end{bmatrix}. \tag{5}$$

Here, the matrix $\boldsymbol{B}(\boldsymbol{s}^{n-1})$ is block diagonal, where the block corresponding to a cell E is $(\lambda(s_E^{n-1})\boldsymbol{T}_E)^{-1}$ with an extended diagonal entry $(\lambda(s_E^{n-1})WI_E^w)^{-1}$ for every well perforating cell E. The matrix $\boldsymbol{C}$ is also block diagonal with each block a $n_E \times 1$ vector of ones, where n_E is the number of faces plus the number well perforations in cell E. Finally, each column of $\boldsymbol{D}$ corresponds to a unique face or well-face and has unit entries in the positions of the face/well-face in the cell-wise ordering. This means that for boundary grid faces, the corresponding column of $\boldsymbol{D}$ has one unit entry, for interior grid faces two unit entries,

and for well-faces the number of unit entries is equal to the number of well-perforations. The first row of the system Eq. 5 are just the collected equations Eqs. 3 and 4, the second row ensures that the total out-flux of every cell is zero, while the third row ensures continuity in the inter-cell fluxes and forces Neumann boundary conditions. Note that this means that for a well face, the corresponding row of $\boldsymbol{D}^{\mathsf{T}}$ sums all the well-perforation rates up to the total rate, which is set as a Neumann boundary condition on the right-hand side. To account for pressure-constraint wells (Dirichlet conditions), we split the vector $\boldsymbol{\pi}^{n\mathsf{T}} = \begin{bmatrix}\widehat{\boldsymbol{\pi}}^{n\mathsf{T}} & \boldsymbol{\pi}_D^{n\,\mathsf{T}}\end{bmatrix}$, where $\boldsymbol{\pi}_D^n$ denotes the (known) pressure at Dirichlet faces and $\widehat{\boldsymbol{\pi}}^n$ the (unknown) interior and Neumann boundary faces. Similarly we split $\boldsymbol{D} = \begin{bmatrix}\widehat{\boldsymbol{D}} & \boldsymbol{D}_D\end{bmatrix}$, and the system becomes

$$\begin{bmatrix} \boldsymbol{B}(\boldsymbol{s}^{n-1}) & \boldsymbol{C} & \widehat{\boldsymbol{D}} \\ \boldsymbol{C}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \\ \widehat{\boldsymbol{D}}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^n \\ -\boldsymbol{p}^n \\ \widehat{\boldsymbol{\pi}}^n \end{bmatrix} = \begin{bmatrix} -\boldsymbol{D}_{\boldsymbol{D}}\boldsymbol{\pi}_D^n \\ \boldsymbol{0} \\ \widehat{\boldsymbol{v}}_N^n \end{bmatrix}. \tag{6}$$

Finally, if we let $\boldsymbol{u}^n$ denote a control input-vector of pressures and/or rates for a given set of the wells at time step t^n, then the two nonzero components of the right-hand side of Eq. 6 may be expressed

$$-\boldsymbol{D}_{\boldsymbol{D}}\boldsymbol{\pi}_D^n = \boldsymbol{A}_D^n\boldsymbol{u} + \boldsymbol{b}_D^n \quad \text{and} \quad \widehat{\boldsymbol{v}}_N^n = \boldsymbol{A}_N^n\boldsymbol{u} + \boldsymbol{b}_N^n, \tag{7}$$

for some matrices $\boldsymbol{A}_D^n$, $\boldsymbol{A}_N^n$, and vectors $\boldsymbol{b}_D^n$, $\boldsymbol{b}_N^n$.

Discretization of the Saturation Equation. For the saturation equation given by Eq. 2 we use a standard upstream weighted implicit finite volume method of the form

$$\boldsymbol{s}^n = \boldsymbol{s}^{n-1} + \Delta t^n \boldsymbol{D}_{PV}^{-1}\left(\boldsymbol{A}(\boldsymbol{v}^n)f(\boldsymbol{s}^n) + \boldsymbol{q}(\boldsymbol{v}^n)_+\right). \tag{8}$$

Here, Δt^n is the time step, $\boldsymbol{D}_{PV}$ is the diagonal matrix containing the cell pore volumes, f is the water fractional-flow function, and $\boldsymbol{q}(\boldsymbol{v}^n)_+$ is the vector of positive cell sources (injection rates). Finally, $\boldsymbol{A}(\boldsymbol{v}^n)$ is the sparse flux matrix with entries $\boldsymbol{A}_{ij} = v$ if cell number j is an upstream neighbor to cell i with flux v, and diagonal entries $\boldsymbol{A}_{ii}$ equal to minus the sum of all fluxes from cell i to downstream neighbors and to production wells that perforate cell i.

Multiscale Mixed Finite-Elements and Coarse Grid Saturation Solver

In this section, we briefly review the current multiscale mixed-finite element formulation (Aarnes et al. 2008) with wells implemented as in Skaflestad and Krogstad (2008), and describe the coarsening of the saturation equation by Aarnes et al. (2007).

The MsMFE Formulation for the Pressure Equation. In the current MsMFE formulation, we consider two grids: a fine grid on which the porosities and permeabilities are given as piecewise constants in each cell, and a coarsened simulation grid, where each block Ω_i consists of a connected set of cells from the underlying fine grid. The approximation spaces for the MsMFE method consist of a constant approximation of the pressure on each coarse block, and a set of velocity basis functions associated with each interface between two blocks and the boundary face between a well and a block. The goal is to approximate the fine grid flux as a linear combination of such basis functions. Consider two neighboring blocks Ω_i and Ω_j, and let Ω_{ij} be a neighborhood containing Ω_i and Ω_j. The basis function $\vec{\psi}_{ij}$ is constructed by numerically solving

$$\vec{\psi}_{ij} = -\lambda(s^0)\mathbf{K}\nabla p_{ij}, \qquad \nabla\cdot\vec{\psi}_{ij} = \begin{cases} w_i(x), & \text{if } x\in\Omega_i, \\ -w_j(x), & \text{if } x\in\Omega_j, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

in Ω_{ij} with $\vec{\psi}_{ij}\cdot\vec{n} = 0$ on $\partial\Omega_{ij}$. Here, $\lambda(s^0)$ is the total mobility for the initial saturation field s^0. If $\Omega_{ij}\neq\Omega_i\cup\Omega_j$, we say that the basis function is computed using overlap or oversampling. The purpose of the weighting function $w_i(x)$ is to produce a flow with unit average over the interface $\partial\Omega_i\cap\partial\Omega_j$ and the function is therefore normalized such that its integral over Ω_i equals one. We refer to the section on numerical experiments for the choice of weighting functions used here.

Next, we discuss basis functions associated with well-block interfaces. Assume that block Ω_i is perforated by well w with boundary γ_w, let Ω_i^w be a neighborhood containing Ω_i, and define $\gamma_i^w = \gamma_w\cap\partial\Omega_i^w$. The basis function $\vec{\psi}_i^w$ is constructed by solving

$$\vec{\psi}_i^w = -\lambda(s^0)\mathbf{K}\nabla p_i^w, \qquad \nabla\cdot\vec{\psi}_i^w = \begin{cases} -w_i(x), & \text{if } x\in\Omega_i, \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

in Ω_i^w with constant p_i^w on γ_i^w and $\vec{\psi}_i^w\cdot\vec{n} = 0$ on $\partial\Omega_i^w\setminus\gamma_i^w$.

Now, let $\boldsymbol{\psi}_{ij}$ and $\boldsymbol{\psi}_i^w$ denote the basis functions resulting from solving Eqs. 9 and 10, respectively. The basis functions are represented as a vector of fluxes (and well rates) analogous the fine-grid flux field $\boldsymbol{v}$ in Eq. 5. Thus the vectors representing the

basis functions are sparse. To generate the multiscale hybrid system, we split the block-block basis functions as $\boldsymbol{\psi}_{ij} = \boldsymbol{\psi}_{ij}^H - \boldsymbol{\psi}_{ji}^H$ such that $\boldsymbol{\psi}_{ij}^H(E)$ is equal $\boldsymbol{\psi}_{ij}(E)$ if $E \in \Omega_{ij} \setminus \Omega_j$ and zero otherwise, and $\boldsymbol{\psi}_{ji}^H(E)$ is equal $-\boldsymbol{\psi}_{ij}(E)$ if $E \in \Omega_j$ and zero otherwise. Next, we arrange all the *hybrid* basis functions $\boldsymbol{\psi}_{ij}^H$ and $\boldsymbol{\psi}_i^k$ block-wise as columns in a matrix $\boldsymbol{\Psi}$. The multiscale pressure system now takes the same form as Eq. 5, where $\boldsymbol{C}$ and $\widehat{\boldsymbol{D}}$ are constructed based on the coarse grid, and the right-hand side is based on the coarse faces :

$$\begin{bmatrix} \boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{B}_f(\boldsymbol{s}_f^{n-1})\boldsymbol{\Psi} & \boldsymbol{C} & \widehat{\boldsymbol{D}} \\ \boldsymbol{C}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \\ \widehat{\boldsymbol{D}}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^n \\ -\boldsymbol{p}^n \\ \widehat{\boldsymbol{\pi}}^n \end{bmatrix} = \begin{bmatrix} -\boldsymbol{D}_{\boldsymbol{D}}\boldsymbol{\pi}_D^n \\ \boldsymbol{0} \\ \widehat{\boldsymbol{v}}_N^n \end{bmatrix}. \tag{11}$$

Here $\boldsymbol{B}_f(\boldsymbol{s}_f^{n-1})$ is the fine-grid matrix based on the fine-grid saturations at the previous time step. By solving the multiscale pressure equation, one obtains coarse-grid fluxes and perforation well rates $\boldsymbol{v}^n$, coarse-grid block pressures $\boldsymbol{p}^n$, and coarse-grid face pressures $\boldsymbol{\pi}^n$. The approximation to the fine-grid flux field is simply given as $\boldsymbol{v}_f^n = \boldsymbol{\Psi}\boldsymbol{v}^n$. In the case when the basis functions are computed without overlap, the matrix $\boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{B}_f(\boldsymbol{s}_f^{n-1})\boldsymbol{\Psi}$ in Eq. 11 becomes block diagonal, and a Schur complement reduction can be applied. When overlap is used, however, we reduce the hybrid system to a mixed system, which is not positive definite, but fast to solve because of the small dimension. We note that in the current approach, the basis functions are computed only once (initially), and thus the computational complexity in solving the pressure equation with the multiscale method is comparable to that of a flow-based upscaling.

Flow-Based Non-Uniform Coarsening of the Saturation Equation. Here we briefly describe the coarsening of the discretized saturation equation suggested by Aarnes et al. (2007) which we employ in this work. The idea is to generate a coarse grid which separates high and low flow regions and at the same time is balanced with respect to gridblock size and the total amount of flow through each coarse block. The grid is constructed based on a single fine-grid flow field, and the overall methodology is shown to be robust both with respect to the level of coarsening and changing well configurations. We refer to Aarnes et al. (2007) for further details, and assume for now a coarse *saturation grid* is given. Let $\boldsymbol{\mathcal{I}}$ denote the prolongation from the coarse saturation grid to the fine grid, such that $\boldsymbol{\mathcal{I}}_{ij}$ is equal to one if block number j contains cell number i. Moreover, define the restriction $\boldsymbol{\mathcal{J}}^{\mathsf{T}} = \boldsymbol{D}_{PV}^{-1}\boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{D}_{PV,f}$, which maps fine-grid saturations to coarse-grid saturations. Here $\boldsymbol{D}_{PV,f}$ denotes the diagonal matrix of pore volumes for the fine-grid as in Eq. 8, and $\boldsymbol{D}_{PV} = \boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{D}_{PV,f}\boldsymbol{\mathcal{I}}$ is the diagonal matrix of pore volumes for the coarse saturation grid. Let $\boldsymbol{s}_f$ denote the fine grid saturations. By multiplying Eq. 8 by $\boldsymbol{\mathcal{J}}^{\mathsf{T}}$ from left, one obtains an expression for the coarse-grid saturations $\boldsymbol{s}$:

$$\boldsymbol{s}^n = \boldsymbol{s}^{n-1} + \Delta t\boldsymbol{D}_{PV}^{-1}\left(\boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{A}(\boldsymbol{v}_f^n)f(\boldsymbol{s}_f^n) + \boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{q}_+\right). \tag{12}$$

Finally, by using the approximation $\boldsymbol{s}_f^n = \boldsymbol{\mathcal{I}}\boldsymbol{s}^n$, the coarse grid saturation scheme reads

$$\boldsymbol{s}^n = \boldsymbol{s}^{n-1} + \Delta t\boldsymbol{D}_{PV}^{-1}\left(\boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{A}(\boldsymbol{v}_f^n)\boldsymbol{\mathcal{I}}f(\boldsymbol{s}^n) + \boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{q}_+\right). \tag{13}$$

We note that Eq. 13 is just an algebraic expression for the saturation scheme, and that the construction of the fine-grid matrix $\boldsymbol{A}(\boldsymbol{v}_f^n)$ is not needed since $\boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{A}(\boldsymbol{v}_f^n)\boldsymbol{\mathcal{I}}$ only depends on the fine scale fluxes over the coarse-grid interfaces.

We end this section by noting that fine-grid quantities are present in both the multiscale formulation for pressure and the implicit finite-volume scheme for the coarse grid saturation. When the two methodologies are combined, this results in exchanging $\boldsymbol{s}_f^{n-1}$ in Eq. 11 with $\boldsymbol{\mathcal{I}}\boldsymbol{s}^{n-1}$, and $\boldsymbol{v}_f^n$ in Eq. 13 by $\boldsymbol{\Psi}\boldsymbol{v}^n$.

Adjoint Formulation

We here briefly review the adjoint formulation, and describe in more detail the adjoint equations for the discretizations considered here. Let $\boldsymbol{x}^n$ denote the state variables at time step n, in our setting this means $\boldsymbol{v}^n$, $\boldsymbol{p}^n$, $\boldsymbol{\pi}^n$, and $\boldsymbol{s}^n$. Moreover, let $\boldsymbol{u}^n$ denote a control input variables (pressure and rate constraints in this setting) at time step n. For each time step we write the equations (e.g., Eqs. 6 and 8) in compact form

$$F^n(\boldsymbol{x}^n, \boldsymbol{x}^{n-1}, \boldsymbol{u}^n), \quad n = 1, \ldots, N, \tag{14}$$

and if we denote by $\boldsymbol{x}$ and $\boldsymbol{u}$ the stacked state and control inputs for all time steps, we express all the equations in the forward simulation by $F(\boldsymbol{x}, \boldsymbol{u}) = 0$. Let $J(\boldsymbol{x}, \boldsymbol{u}) = \sum_{n=1}^N J^n(\boldsymbol{x}^n, \boldsymbol{u}^n)$ be an objective function and denote by $\nabla_{\boldsymbol{u}}J$ the gradient with respect to $\boldsymbol{u}$. Obtaining the gradient directly is not feasible since this would require computing the matrix $\frac{d\boldsymbol{x}}{d\boldsymbol{u}}$, but one gets around this by introducing the auxiliary function J_λ:

$$J_\lambda(\boldsymbol{x}, \boldsymbol{u}) = J(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^{\mathsf{T}}F(\boldsymbol{x}, \boldsymbol{u}), \tag{15}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers. The gradient $\nabla_{\boldsymbol{u}}J_\lambda$ is now given as

$$(\nabla_{\boldsymbol{u}}J_\lambda)^{\mathsf{T}} = \frac{\partial J}{\partial \boldsymbol{u}} + \frac{\partial J}{\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{d\boldsymbol{u}} + \boldsymbol{\lambda}^{\mathsf{T}}\frac{\partial F}{\partial \boldsymbol{u}} + \boldsymbol{\lambda}^{\mathsf{T}}\frac{\partial F}{\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{d\boldsymbol{u}} + F^{\mathsf{T}}\frac{d\boldsymbol{\lambda}}{d\boldsymbol{u}}, \tag{16}$$

which reduces to

$$(\nabla_{\boldsymbol{u}} J_\lambda)^{\mathsf{T}} = \frac{\partial J}{\partial \boldsymbol{u}} + \boldsymbol{\lambda}^{\mathsf{T}} \frac{\partial F}{\partial \boldsymbol{u}}, \tag{17}$$

when $\boldsymbol{\lambda}$ obeys the *adjoint equations*

$$\frac{\partial F}{\partial \boldsymbol{x}}^{\mathsf{T}} \boldsymbol{\lambda} = -\frac{\partial J}{\partial \boldsymbol{x}}^{\mathsf{T}}. \tag{18}$$

We will now describe the adjoint equations for the fine and coarse grid sequential discretizations employed in this paper in greater detail.

Fine Grid Adjoint Equations. For the derivation of the adjoint equations corresponding to Eqs. 6 and 8, we introduce Lagrange multipliers $\boldsymbol{\lambda}_u^n$,$\boldsymbol{\lambda}_p^n$,$\boldsymbol{\lambda}_{\widehat{\pi}}^n$, and $\boldsymbol{\lambda}_s^n$ for each time step t^n corresponding to the dual variables $\boldsymbol{v}^n$, $\boldsymbol{p}^n$, $\widehat{\boldsymbol{\pi}}^n$, and $\boldsymbol{s}^n$, respectively. For ease of notation let $g(\boldsymbol{v}, \boldsymbol{s}) = \Delta t \boldsymbol{D}_{PV}^{-1} \left(\boldsymbol{A}(\boldsymbol{v}) f(\boldsymbol{s}) + \boldsymbol{q}(\boldsymbol{v})_+ \right)$. Assuming the objective function J is a function of fluxes (including rates), saturations, and control input variables, the adjoint equations for time step n becomes

$$\left(\boldsymbol{I} - \frac{\partial g(\boldsymbol{v}^n, \boldsymbol{s}^n)}{\partial \boldsymbol{s}^n} \right)^{\mathsf{T}} \boldsymbol{\lambda}_s^n = \boldsymbol{\lambda}_s^{n+1} - \frac{\partial J^n}{\partial \boldsymbol{s}^n}^{\mathsf{T}} - \left(\frac{\partial \boldsymbol{B}(\boldsymbol{s}^n) \boldsymbol{v}^{n+1}}{\partial \boldsymbol{s}^n} \right)^{\mathsf{T}} \boldsymbol{\lambda}_u^{n+1}, \tag{19}$$

$$\begin{bmatrix} \boldsymbol{B}(\boldsymbol{s}^{n-1}) & \boldsymbol{C} & \widehat{\boldsymbol{D}} \\ \boldsymbol{C}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \\ \widehat{\boldsymbol{D}}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_u^n \\ \boldsymbol{\lambda}_p^n \\ \boldsymbol{\lambda}_{\widehat{\pi}}^n \end{bmatrix} = \begin{bmatrix} -\frac{\partial J^n}{\partial \boldsymbol{v}^n}^{\mathsf{T}} - \frac{\partial g(\boldsymbol{v}^n, \boldsymbol{s}^n)}{\partial \boldsymbol{v}^n}^{\mathsf{T}} \boldsymbol{\lambda}_s^n \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \tag{20}$$

where any term involving index $M+1$ is neglected and M is the number of time steps. Note that in the linear system for $\boldsymbol{\lambda}_s^n$, $\boldsymbol{\lambda}_s^{n+1}$ appears on the right hand side, so the equations need to be solved backwards in time. In this work we use analytical relative permeabilities, and thus all partial derivatives in Eqs. 19 and 20 are computed analytically. Once the adjoint equations have been solved, the gradient of the objective function J at time step t^n is given as

$$\nabla_{\boldsymbol{u}^n} J = \frac{\partial J}{\partial \boldsymbol{u}^n}^{\mathsf{T}} - \boldsymbol{A}_D^{n\,\mathsf{T}} \boldsymbol{\lambda}_v^n - \boldsymbol{A}_N^{n\,\mathsf{T}} \boldsymbol{\lambda}_{\widehat{\pi}}^n, \tag{21}$$

where $\boldsymbol{A}_D^n$ and $\boldsymbol{A}_N^n$ are the matrices appearing in Eq. 7 representing the control part of the right hand side of the linear system Eq. 6.

Coarse Grid / Multiscale Adjoint Equations. The construction of the adjoint equations for the coarse model is analogous to the forward version. The coarse-model multipliers $\boldsymbol{\lambda}_v^n$ approximate the fine-model multipliers $\boldsymbol{\lambda}_{v,f}^n$ by $\boldsymbol{\lambda}_{v,f}^n \approx \boldsymbol{\Psi} \boldsymbol{\lambda}_v^n$, while $\boldsymbol{\lambda}_s^n$ approximates $\boldsymbol{\lambda}_{s,f}^n$ by $\boldsymbol{\lambda}_{s,f}^n \approx \boldsymbol{\mathcal{J}} \boldsymbol{\lambda}_s^n$ (in contrast to the relation $\boldsymbol{s}_f^n \approx \boldsymbol{\mathcal{I}} \boldsymbol{s}^n$). By inserting these relations in the fine-model version above and summing the equations, one obtains

$$\left(\boldsymbol{I} - \boldsymbol{\mathcal{J}}^{\mathsf{T}} \frac{\partial g(\boldsymbol{v}_f^n, \boldsymbol{s}_f^n)}{\partial \boldsymbol{s}_f^n} \boldsymbol{\mathcal{I}} \right)^{\mathsf{T}} \boldsymbol{\lambda}_s^n = \boldsymbol{\lambda}_s^{n+1} - \left(\frac{\partial J^n}{\partial \boldsymbol{s}_f^n} \boldsymbol{\mathcal{I}} \right)^{\mathsf{T}} - \left(\boldsymbol{\Psi}^{\mathsf{T}} \frac{\partial \boldsymbol{B}(\boldsymbol{s}_f^n) \boldsymbol{v}_f^{n+1}}{\partial \boldsymbol{s}_f^n} \boldsymbol{\mathcal{I}} \right)^{\mathsf{T}} \boldsymbol{\lambda}_u^{n+1}, \tag{22}$$

$$\begin{bmatrix} \boldsymbol{\Psi}^{\mathsf{T}} \boldsymbol{B}_f(\boldsymbol{s}_f^{n-1}) \boldsymbol{\Psi} & \boldsymbol{C} & \widehat{\boldsymbol{D}} \\ \boldsymbol{C}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \\ \widehat{\boldsymbol{D}}^{\mathsf{T}} & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_u^n \\ \boldsymbol{\lambda}_p^n \\ \boldsymbol{\lambda}_{\widehat{\pi}}^n \end{bmatrix} = \begin{bmatrix} -\left(\frac{\partial J^n}{\partial \boldsymbol{v}_f^n} \boldsymbol{\Psi} \right)^{\mathsf{T}} - \left(\boldsymbol{\mathcal{J}}^{\mathsf{T}} \frac{\partial g(\boldsymbol{v}_f^n, \boldsymbol{s}_f^n)}{\partial \boldsymbol{v}_f^n} \boldsymbol{\Psi} \right)^{\mathsf{T}} \boldsymbol{\lambda}_s^n \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \tag{23}$$

with $\boldsymbol{s}_f^i = \boldsymbol{\mathcal{I}} \boldsymbol{s}^i$ and $\boldsymbol{v}_f^i = \boldsymbol{\Psi} \boldsymbol{v}^i$ and where $\boldsymbol{C}$ and $\widehat{\boldsymbol{D}}$ are the same matrices as in the forward multiscale system. The gradient is obtained as in Eq. 21, just exchanging $\boldsymbol{A}_D^n$ and $\boldsymbol{A}_N^n$ by the coarse system analogues.

Reducing the Computational Complexity

For typical reservoir simulators, the main work horse and time consumer is the linear solver. This is also the case for the fine model discretization considered here. For the coarse model, however, the relative amount of time consumed by the linear solver becomes almost insignificant when the level of coarsening is high. For a naive implementation when overlap is used in the computation of basis functions, computing the matrix product $\boldsymbol{\Psi}^{\mathsf{T}} \boldsymbol{B}_f(\boldsymbol{s}_f^{n-1}) \boldsymbol{\Psi}$ in Eqs. 11 and 23 consumes over 90% of the total computation time for a coarsening factor of about 100 in the current implementation. However, since the saturation here is solved on a fixed coarse grid, one can bypass the costly matrix products involving the fine model $\boldsymbol{B}_f$ during the simulation. Consider the relation

$$\boldsymbol{\Psi}^{\mathsf{T}} \boldsymbol{B}_f(\boldsymbol{\mathcal{I}} \boldsymbol{s}^{n-1}) \boldsymbol{\Psi} = \sum_{k=1}^{N_s} \boldsymbol{\Psi}^{\mathsf{T}} \boldsymbol{B}_f(\boldsymbol{\mathcal{I}} \boldsymbol{e}_k s_k^{n-1}) \boldsymbol{\Psi}, \tag{24}$$

where N_s is the number of blocks in the coarse saturation grid and $\boldsymbol{e}_k$ is the k-th unit vector. The summands in Eq. 24 become time independent if multiplied by $\lambda(s_k^{n-1})$ and thus the matrices $\lambda(s_k^{n-1})\boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{B}_f(\boldsymbol{\mathcal{I}}\boldsymbol{e}_k s_k^{n-1})\boldsymbol{\Psi}$ can be computed in advance as a preprocessing step. In the current implementation, they are represented as sparse matrix input (indices and values). Additional computational savings can be obtained when considering the coarse saturation discretization in Eq. 13 and the adjoint counterpart in Eq. 22. As mentioned earlier, the construction of the matrix $\boldsymbol{\mathcal{I}}^{\mathsf{T}}\boldsymbol{A}(\boldsymbol{v}_f^n)\boldsymbol{\mathcal{I}}$ does not require the full fine-grid flux field $\boldsymbol{v}_f^n$, but only the fine-grid fluxes over the coarse-grid faces. Accordingly, one does not need to do computations with the full matrix $\boldsymbol{\Psi}$ of basis functions during simulations, only with the rows corresponding to fine-grid faces lying on coarse-grid faces. We will report on the computational speed-up of applying these techniques in the next section.

Numerical Experiments.

In this section we present two numerical examples; in the first we consider optimizing rates on several 2D Cartesian models. In the second we optimize net-present value for a real reservoir model with artificial wells using various levels of coarsening.

For improved accuracy, we use oversampling when computing the multiscale basis functions, enlarging the support domain of each basis by half the block diameter in all directions. The weighting functions used in the computation of basis functions are as in previous publications (Aarnes et al. 2006) given by

$$w_i(x) = \frac{\operatorname{trace}(\mathbf{K})}{\int_{\Omega_i} \operatorname{trace}(\mathbf{K})},$$

where trace denotes the sum of the diagonal elements. We stress the point that all fine-grid computations are performed as a preprocessing, so the computation of the multiscale basis functions and the generation of the coarse saturation grid is performed only once. The preprocessing steps can be expressed as follows:

1. Partition fine grid into a logically Cartesian coarse pressure grid, followed by a processing routine that makes sure all coarse blocks are connected.

2. Compute multiscale basis functions based on the coarse pressure grid and initial fine grid saturation.

3. Solve one single multiscale pressure equation using initial (base case) rates and pressures to obtain a fine grid velocity field. Based on this velocity field, construct the coarse non-uniform saturation grid.

4. Compute the sparse matrix input indices and values of the summands in Eq. 24 and reduce the basis matrix $\boldsymbol{\Psi}$ according to the coarse mappings between the pressure and saturation discretizations. After this step, all fine-grid information can be stored and cleared from memory.

After the preprocessing steps, the optimization process is run. This consists of the forward and adjoint simulations to obtain a gradient, and performing a line search along the (projected) gradient. When linear equality constraints are present we perform an orthogonal projection of the gradient onto the space spanned by the constraint and restrict the step size according to the linear inequality constraints. In these examples, we do not consider nonlinear constraints, although the approaches of Sarma et al. (2008) or Kraaijevanger et al. (2007) could also be employed here.

Optimizing Recovery on 2D Heterogeneous Models. In this example, we consider 2D layers sampled from the Tenth SPE Comparative Solution Project (Christie and Blunt 2001). Thus, the fine grid consists of 60×220 grid cells each of size 20×10 feet. We use a quarter five-spot well configuration with one injector at cell $(30, 110)$, and four producers in the corner cells. The center well injects water at constant rate for 0.4 PVI (pore volumes injected), and the four corner wells produce at equal rate in the initial configuration. We use quadratic relative permeabilities with a water-to-oil mobility ratio of 5. We attempt to find production rates to optimize the recovery using two coarse models. For the finer coarse model we use a 10×22 grid for pressure, and saturation grids consisting of about 600 cells, and for the coarser coarse model we use a 3×11 grid for pressure and saturation grids consisting of about 150 cells, thus an upscaling factor of 400 for the pressure equation and almost 100 for the saturation equation. After the coarse model optimization process has converged, we continue the optimization using the fine model. In this way, we are able to say something about the distance from local optimum of the coarse model to local optimum of the fine model. In **Fig. 1** we have plotted the results obtained for Layers 26–35 and 76–85. Although the improvement is not large for all the layers, it is clear that both coarse models succeed in giving close to optimal rates for the fine model. As expected, the finer coarse models perform slightly better than the coarser.

We now look in more detail at Layer 76 using the coarsest model. In **Fig. 2**, we have plotted the saturation grid consisting of about 150 cells. The fine and coarse model saturations are depicted for the initial rate configuration and optimal rate settings. It is seen that the optimal rates result in a significant sweep improvement in the upper left part of the model. In **Fig. 3**, the objective values are plotted after each iteration in the optimization process. The coarse model optimization converges after 16 iterations giving a significant improvement in recovery. Note that the solid line depicts the objective function values computed using the fine model, so this line need not be monotonic, in contrast to the dashed line showing the objective values computed using the coarse model. After the 16 iterations, the optimization is continued for a few iterations using the fine model, only to achieve an

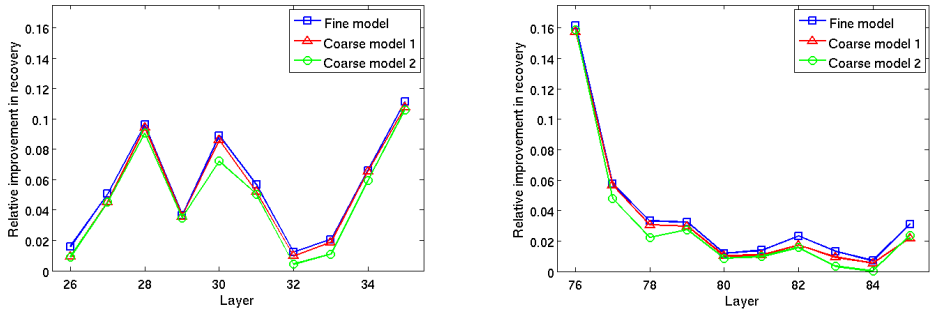

Fig. 1—Plots showing the improvements obtained using the coarse models for the various layers followed by an optimization using the fine model. Layers 26–35 have log-normally permeability fields, while Layers 76–85 contain high-permeable channels.

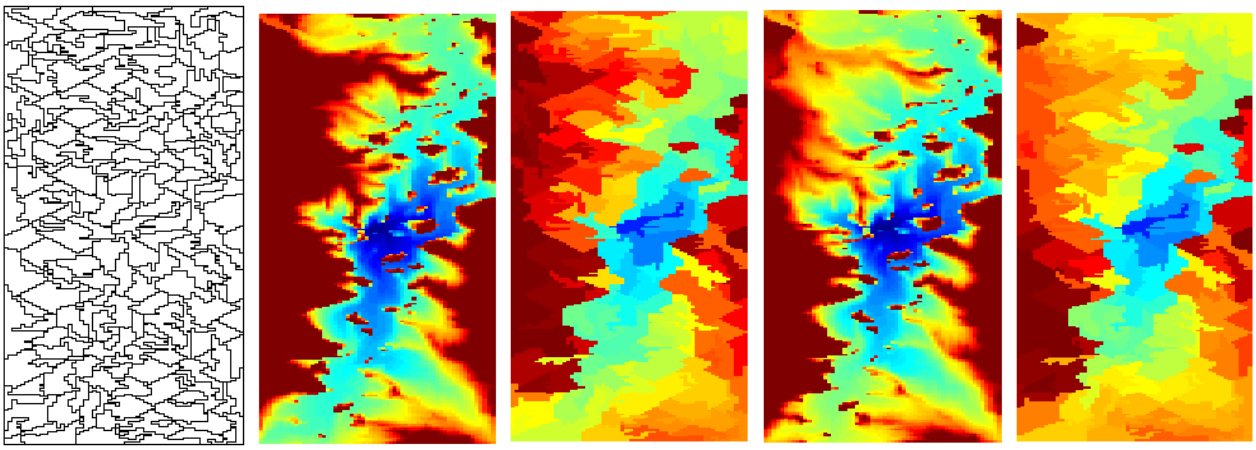

Fig. 2—The left plot depicts the coarse-model saturation grid for Layer 76. The four rightmost plots show end-time saturation for initial configuration fine model, initial configuration coarse model, optimal configuration fine model, and optimal configuration coarse model .

insignificant improvement in objective value. The right plot in Fig. 3 shows the rate settings after the coarse model optimization (dashed line) and after the fine model optimization (solid line).

Optimizing Net-Present Value on a Real Field Model Geometry. In this example we consider a simulation model of a real field containing about $45,000$ grid cells. The permeability is mainly layered and ranges four orders of magnitude. The model has several faults, but we here neglect fault multipliers in computations. For sake of the example, we assume the reservoir is initially filled with oil, and position seven production wells and four production wells more or less arbitrarily as shown in **Fig. 4**. All the wells are vertical and perforate every cell from top to bottom. Again, we use quadratic relative permeabilities with a water-to-oil mobility ratio of 5. Initially, all producers produce at equal rate and all injectors inject at equal rate for a total of 1 PVI. We attempt to optimize net-present value (NPV), for which the water injection cost and the water production cost is set to 10% and 15% of the oil revenue, respectively, and the discount factor is neglected. The total injection rate is allowed to vary as our only equality constraint is that the sum of all rates (positive and negative) should equal zero. The only inequality constraints are that injectors are not allowed to switch to producers and vice versa.

In this example we consider six coarse models: two coarse pressure grids consisting of 82 and 365 grid blocks resulting from initial partitionings in index space of sizes $4 \times 9 \times 2$ and $7 \times 15 \times 4$, respectively, and three coarse saturation grids consisting of 136, 291 and 800 blocks. We refer to these models as Model 1 to 6, where Model 1, 2 and 3 use the coarser pressure grid and saturation grids consisting of 136, 291, and 800 blocks, respectively, and Models 4 to 6 are the remaining combinations in similar order. In **Fig. 5**, the coarse pressure grid and saturation grid for Model 1 are plotted with blocks colored randomly. To avoid huge jumps in the rates over time, we initially run ten iterations keeping the well rates constant in time, and then introduce 20 control steps in time and iterate until convergence. Initial tests showed that this approach also improved the value of the objection function. In **Fig. 6**, we compare the coarse and fine model values throughout the iteration process. Again, we plot the values computed using both the coarse model and the fine model. As seen, all the models give rates giving NPV close to that of the fine-model optimization, and two of the models even produce higher values. We also note that the optimization converges faster

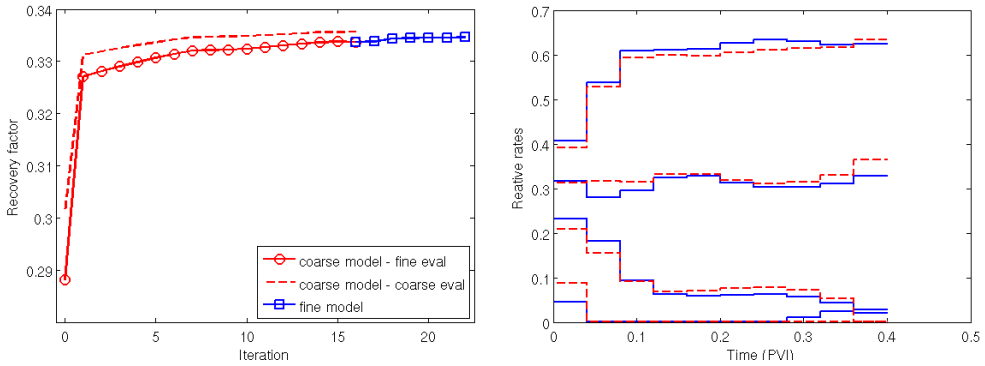

Fig. 3—Left plto depicts objective values obtained during the optimization process. The coarse model is used for 16 iterations, followed by 6 iterations using the fine model. The solid lines show the objective function values computed using the fine model, and the dashed line show the values computed using the coarse model. Right plot shows well rates for coarse-model (dashed lines) and fine-model (solid lines) optmimum, respectively.

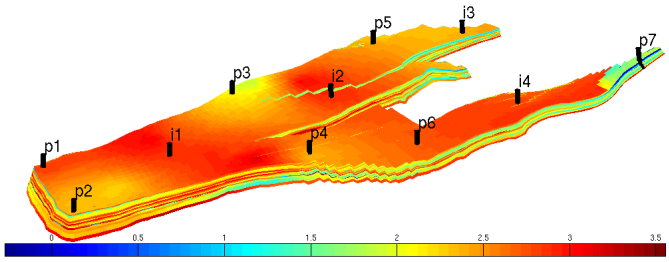

Fig. 4—Model used in numerical example; $\log_{10}$ of permeability field and wells used for simulation.

for all the coarse models, and after 10 iterations when the rates are no longer forced to be constant over time, little improvement is observed. It is also interesting to note that all the models give optimal settings for which Producers 5 and 6 and Injector 1 produce/inject next to nothing. Even though the obtained NPV for the various models match closely, the optimal rates vary substantially. For instance, the total amount of injected water is 0.94 PVI for the fine model, while ranging from 0.77 to 1.02 PVI for the coarse models.

The computations for this example were performed on a standard workstation, which spent the night on the optimization based on the fine model. For the coarser models each forward simulation (using 20 time steps) took from approximately 5 to 20 seconds when the sparse matrix inputs were computed during the preprocessing as described in the previous section. For the coarsest model, the preprocessing resulted in a speedup factor of 10 or higher compared with running the simulation computing all matrix products of the form $\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{B}_f\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}\boldsymbol{v}$ directly. The preprocessing steps took between 4 and 8 minutes for the various coarse models. Finally, we note that this is a prototype MATLAB implementation, and that further computational savings could be obtained. As a result, the overall optimization process based on the coarsest model considered here should be able to run in a couple of minutes on a laptop.

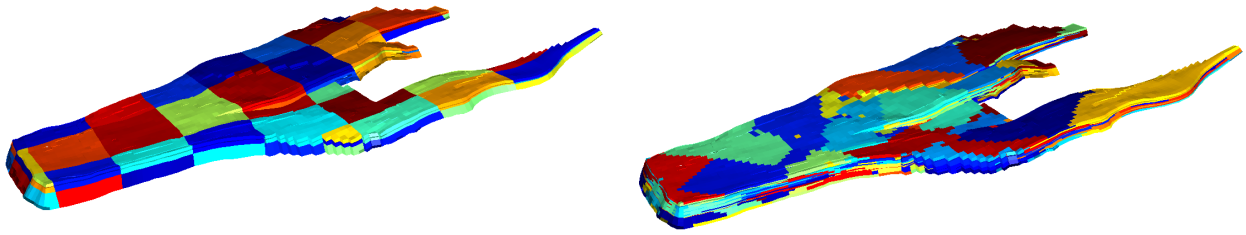

Fig. 5—Coarse pressure grid (left) and coarse saturation grid (right).

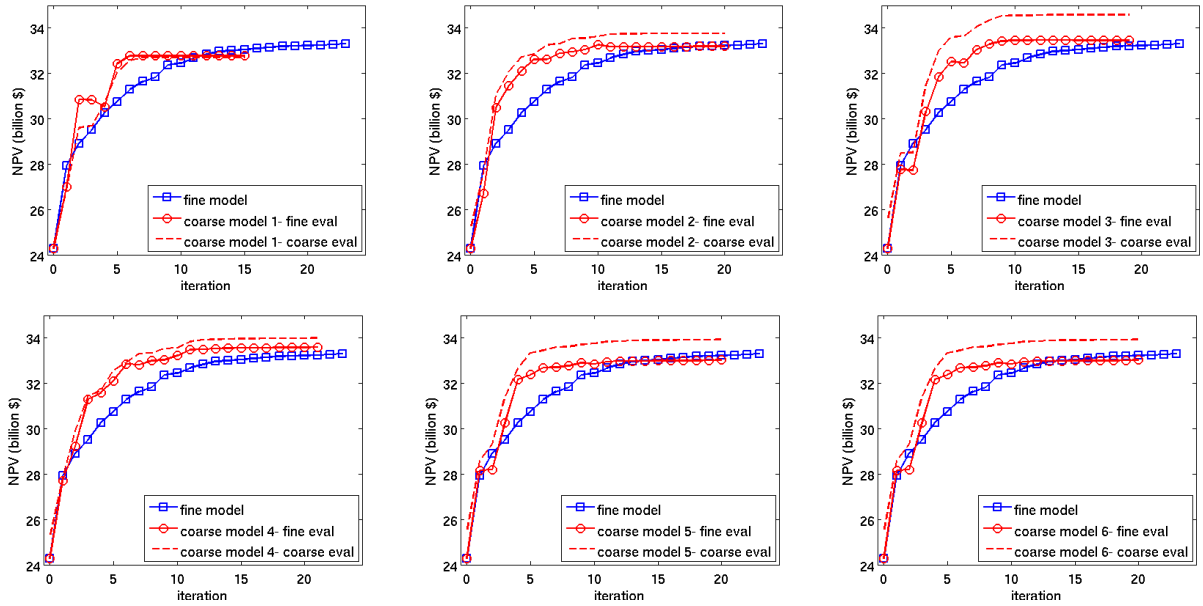

Fig. 6—Objective values obtained with the various coarse models compared with the fine-model optimization. The coarsest pressure grid is used in top row, while coarsest saturation grid is used in the left column.

Conclusions

In this paper we have considered a coarse adjoint model for two-phase incompressible flow. The forward model consists of a mixed mulitscale pressure solver coupled with a flow-based coarse-grid saturation solver. For improved computational efficiency, we employ preprocessing steps which compute coarse mappings between the discretized pressure and saturation equations, and obtain an order of magnitude speed-up. As a result, we are able to optimize the water flooding on a 45,000 grid-cell model within minutes.

The suggested methodology as a model reduction tool has the advantage that the multiscale basis functions are local, and thus completely process independent. We therefore foresee that the methodology has a great potential also in well-placement problems as introducing/moving a well only requires adding a few basis functions based on local computations. This will be addressed in future research.

Acknowledgements

The authors gratefully acknowledge financial support by the Center for Intgrated Operations in the Petroleum Industry at NTNU (Krogstad) and from the Research Council of Norway under grants number 174551/S30 (Hauge and Gulbransen) and 186935/I30.

Nomenclature

Physical quantities:

f = fractional flow function
$\mathbf{K}$ = absolute permeability
p = pressure
s = saturation
t = time
$\vec{v}$ = total Darcy velocity
q = volumetric rate
WI = well index
λ = total mobility
ϕ = porosity
π = face pressure

Domain and grid:

Ω = entire physical domain
∂D = boundary of domain D
E = cell in the fine grid
Ω_i = coarse block number i
Ω_{ij} = support for basis function $\vec{\psi}_{ij}$
Ω_i^w = support for basis function $\vec{\psi}_i^w$
γ_w = boundary of well w
γ_i^w = interface between well w and Ω_i^k

Functions, etc:

F = reservoir equations
J = objective function
$\vec{\psi}_{ij}$ = basis function, interface of block i and j
$\vec{\psi}_i^w$ = basis function, interface of block i and well w
w_i = weight function associated with coarse block Ω_i

Vectors and matrices:

$\boldsymbol{s}$ = vector of cell/block saturations
$\boldsymbol{p}$ = vector of cell/block pressures
$\boldsymbol{\pi}$ = vector of cell/block face pressures
$\boldsymbol{v}$ = vector of outward fluxes on cell/block faces
$\boldsymbol{B}$ = inner product matrix of velocity basis functions
$\boldsymbol{C}$ = integral of the divergence of velocity b.f.
$\boldsymbol{\mathcal{I}}$ = coarse to fine grid prolongation
$\boldsymbol{\mathcal{J}}^{\mathsf{T}}$ = fine to coarse grid restriction
$\boldsymbol{T}_E$ = transmissibility matrix for cell E
$\boldsymbol{\Psi}$ = matrix of all basis functions
$\boldsymbol{\lambda}_\alpha$ = Lagrange multiplier corresponding to state α

Numbers:

N = number of cells in fine grid
N_p = number blocks in coarse pressure grid
N_s = number blocks in coarse saturation grid
M = number of time steps
n_E = number of faces in cell E

Subscripts:

i, j = block/cell numbers
f = fine grid/model quantity
w = well number
N = Neumann/flux boundary
D = Dirichlet/pressure boundary

Superscripts:

w = well number
n = time step
H = hybrid formualation

References

Aarnes, J. E., Hauge, V. L., and Efendiev, Y. 2007. Coarsening of Three-Dimensional Structured and Unstructured Grids for Subsurface Flow. *Adv. Water Resour.*, 30(11):2177–2193.

Aarnes, J. E., Krogstad, S., and Lie, K.-A. 2006. A Hierarchical Multiscale Method for Two-Phase Flow Based Upon Mixed Finite Elements and Nonuniform Coarse Grids. *Multiscale Model. Simul.*, 5(2):337–363 (electronic).

Aarnes, J. E., Krogstad, S., and Lie, K.-A. 2008. Multiscale Mixed/Mimetic Methods on Corner-Point Grids. *Comput. Geosci.*, 12(3):297–315. Doi:10.1007/s10596-007-9072-8.

Asheim, H. 1987. Optimal Control of Water Drive. Paper SPE 15978.

Aziz, K. and Settari, A. 1979. *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers, London and New York.

Brezzi, F. and Fortin, M. 1991. *Mixed and Hybrid Finite Element Methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York.

Brezzi, F., Lipnikov, K., and Simoncini, V. 2005. A Family of Mimetic Finite Difference Methods on Polygonial and Polyhedral Meshes. *Math. Models Methods Appl. Sci.*, 15:1533–1553.

Brouwer, D. R. and Jansen, J. D. 2004. Dynamic Optimization of Waterflooding with Smart Wells Using Optimal Control Theory. *SPE J.*, 9(4):391–402. Paper SPE 78278.

Cardoso, M. A., Durlofsky, L. J., and Sarma, P. 2008. Development and Application of Reduced-Order Modeling Procedures for Subsurface Flow Simulation. *Int. J. Numer. Meth. Eng., to appear.*

Chen, Z. and Hou, T. 2003. A Mixed Multiscale Finite Element Method for Elliptic Problems With Oscillating Coefficients. *Math. Comp.*, 72:541–576.

Christie, M. A. and Blunt, M. J. 2001. Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. *SPE Reservoir Eval. Eng.*, 4:308–317. Url: `http://www.spe.org/csp/`.

Hou, T. and Wu, X.-H. 1997. A Multiscale Finite Element Method for Elliptic Problems in Composite Materials and Porous Media. *J. Comput. Phys.*, 134:169–189.

Jenny, P., Lee, S. H., and Tchelepi, H. A. 2003. Multi-Scale Finite-Volume Method for Elliptic Problems in Subsurface Flow Simulation. *J. Comput. Phys.*, 187:47–67.

Kraaijevanger, J. F. B. M., Egberts, P. J. P., Valstar, J. R., and Buurman, H. W. 2007. Optimal Waterflood Design Using the Adjoint Method. Paper SPE 105764 presented at 2007 SPE RRS, Houston, TX, U.S.A, 26–29 February.

Krogstad, S., Lie, K.-A., Nilsen, H. M., Natvig, J. R., Skaflestad, B., and Aarnes, J. E. 2009. A Multiscale Mixed Finite-Element Solver for Three-Phase Black-Oil Flow. Paper SPE 118993 presented at 2009 SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February.

Li, R., Reynolds, A., and Oliver, D. 2003. History Matching of Three-Phase Flow Production Data. *SPE J.*, 8(4):328–340. Paper SPE 87336.

Peaceman, D. 1983. Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability. *SPE J.*, 23(3):531–543. Paper SPE 10528.

Ramirez, W. F. 1987. *Application of Optimal Control Theory to Enhanced Oil Recovery (Developments in Petroleum Science)*. Elsevier Science Publishers.

Sarma, P., Chen, W. H., Durlofsky, L. J., and Aziz, K. 2008. Production Optimization with Adjoint Models Under Nonlinear Control-State Path Inequality Constraints. *SPEREE*, 11(2):326–339. Paper SPE 99959.

Sarma, P., Durlofsky, L., Aziz, K., and Chen, W. 2006. Efficient Real-Time Reservoir Management Using Adjoint-Based Optimal Control and Model Updating. *Comput. Geosci.*, 10(1):3–36.

Skaflestad, B. and Krogstad, S. 2008. Multiscale/Mimetic Pressure Solvers With Near-Well Grid Adaption. In *Proceedings of ECMOR XI–11th European Conference on the Mathematics of Oil Recovery*, number A36, Bergen, Norway. EAGE.

van Doren, J. F. M., Markovinovic, R., and Jansen, J. D. 2006. Reduced-Order Optimal Control of Water Flooding Using Proper Orthogonal Decomposition. *Comput. Geosci.*, 10(1):137–158.