Multiscale-Streamline Simulation and Dynamic Data Integration for High-Resolution Subsurface Models

V.R. Stenerud, V. Kippe, K.–A. Lie, and A. Datta–Gupta

Abstract. We discuss an efficient method for integrating dynamic data in high-resolution subsurface models. The method consists of two key technologies: (i) a very fast multiscale-streamline flow simulator, and (ii) a fast and robust 'generalized travel-time inversion' method. The travel-time inversion is based on sensitivities computed analytically along streamlines using only one forward simulation. The sensitivities are also used to selectively reduce the updating of basis functions in the multiscale mixed finite-element pressure solver. Moreover, we propose a new streamline formulation that improves the accuracy of production curves and allows a drastic reduction in the number of streamlines required to calculate accurate dynamic data responses.

The accuracy and robustness of our method is discussed using two 2-D test cases. Furthermore, We demonstrate the efficiency and utility of our approach using a highly detailed 3-D subsurface model consisting of more than one million cells and 69 producing wells, for which seven years of dynamic data are integrated in less than twenty minutes on a standard workstation PC.

1. Introduction

Subsurface models are usually built based on static date that are either confined in space or have low spatial resolution. Dynamic data must therefore be integrated into the subsurface model in order to give reliable predictions of future dynamic flow responses. In recent years it has become common to formulate the integration of dynamic data as an inverse problem and inversion methods based on a streamline formulation have shown to be particularly promising in this regard. In a recent paper (*Stenerud et al.*, to appear), we introduced a particularly efficient inversion strategy designed especially for integrating dynamic data into high-resolution subsurface models with millions of cells. The strategy consists of two technologies: a generalized travel-time inversion method (*Vasco et al.*, 1999; *He et al.*, 2002) based on sensitivities computed analytically along streamlines and a highly efficient multiscale-streamline flow solver (*Aarnes et al.*, 2005).

The generalized travel-time inversion method (*Vasco et al.*, 1999; *He et al.*, 2002) has previously been successfully applied to many field cases from the petroleum industry (see e.g., *Qassab et al.*, 2003; *Hohl et al.*, 2006). The method was chosen primarily because it is robust, computationally efficient, and tends to conserve geological realism in the inverted model. Unlike conventional amplitude inversion, which can be highly nonlinear, the travel-time inversion has been shown to have quasilinear properties (*Vasco et al.*, 1999; *Cheng et al.*, 2005). The minimization therefore proceeds rapidly even if the initial model is not close to the global minimum, avoids over-corrections near fluid sources and sinks, and does not converge to secondary peaks that are typically seen in dynamic data from real-field cases, e.g., tracer data.

Each iteration in the inversion method must be accompanied by a forward simulation, which typically will be the most time-consuming part of the inversion process. Streamline methods are particularly suitable for simulating flow in large and geologically complex models, where the fluid flow is dictated primarily by heterogeneity in rock properties, positions of fluid sinks/sources, and phase mobilities. In general, streamline simulators have low memory requirements, high computational efficiency, and scale (almost) linearly with model size. Therefore, streamline simulation offers the opportunity to solve outstanding engineering queries that might otherwise be difficult or impossible to address using other approaches. Within the petroleum industry, streamline simulators are progressively being used more by operating companies as an alternative to traditional reservoir simulators in several reservoir engineering workflows, including: screening of enhanced recovery projects, rapid sensitivity studies, history matching, uncertainty assessment, upscaling, flood optimization, or simulation studies of sector or full-field models.

Even though streamline simulation provides fast forward simulation compared with a full finite-difference simulation in 3-D, computing pressure and fluid velocities still remains an expensive part of the inversion algorithm. As a result, the inversion process is therefore usually performed on upscaled subsurface models, although this may result in loss of important fine-scale information. In (*Stenerud et al.*, to appear), we proposed to replace the conventional pressure solver used in current streamline simulators by a much faster multiscale pressure solver (*Aarnes et al.*, 2005). The multiscale solver can be seen as a method that upscales and downscales the flow equations in a single step. In an upscaling method, the fine grid of the subsurface model is coarsened to form a simulation grid, on which the global flow equation is solved. To this end, one typically solves local flow problems inside each (pair of) grid block(s) and computes the effective permeability (or transmissibility) value that preserves this flow in an averaged sense. Similarly, the multiscale mixed finite-element method (MsMFEM) used herein solves a local flow problem for each pair of neighboring grid blocks in the coarse grid and uses the local flow solution as a basic building block (basis function) on the coarse grid. As other multiscale methods, MsMFEM is primarily targeted at dynamic flow simulations, where the pressure needs to be computed repeatedly. High efficiency is achieved since most basis functions can be reused from the previous pressure solve and updating reduces to solving a global equation on the coarse grid. In (*Stenerud et al.*, to appear) we proposed to use sensitivity coefficients to locate basis functions that need to be updated from one pressure solve to the next. The resulting pressure solver is robust and produces mass-conservative flow velocities both on the coarse grid and on the underlying fine grid.

The purpose of the current paper is two-fold: First, we present a modified streamline formulation that allows us to drastically reduce the number of streamlines needed to compute accurate production curves from the flow simulation. Second, we present a more in-depth analysis of the efficiency and robustness of the multiscale-streamline data-integration method, and in particular for the associated strategy for work reduction based on sensitivity coefficients.

To achieve high efficiency in the streamline simulation, it is clearly desirable to use as few streamlines as possible. On the other hand, the set of streamlines should be representative and sufficiently dense to ensure accurate prediction of flow patterns and production responses, and to limit errors in the mass balance. Lack of mass conservation is a problem of particular concern to reservoir engineers, and in this paper we will try to analyze the lack of mass conservation and suggest methodological improvements that will strongly improve the quality of measured production curves (dynamical responses). This will in turn allow a significant reduction in the number of streamlines required to ensure highly accurate production curves.

The outline of the paper is as follows. Section 2 presents the multiscale-streamline solver. In Section 3 we discuss mass-balance errors and present a strategy to obtain accurate production curves using a small number of streamlines. Section 4 presents our approach for data integration and discusses its practical applicability using several numerical examples. Finally, our results are summarized in Section 5.

2. Multiscale-Streamline Simulation

We consider incompressible two-phase flow of oil and water in a non-deformable permeable medium and neglect the effects of gravity, compressibility and capillary forces. Further, we also assume for simplicity no-flow boundary conditions for the reservoir. Our flow model then consists of an elliptic pressure equation

$$\vec{u} = -\lambda_t(S)k\nabla p, \quad \nabla \cdot \vec{u} = q_t. \tag{1}$$

and a quasilinear hyperbolic transport equation

$$\phi\frac{\partial S}{\partial t} + \nabla \cdot (f_w(S)\vec{u}) = q_w \tag{2}$$

The primary unknowns in the coupled system **Eqs. 1** and **2** are the pressure p, the total (Darcy) velocity $\vec{u}$, and the water saturation S. The underlying porous rock formation is modeled in terms of the absolute permeability k and the porosity ϕ, which henceforth are assumed to depend on the spatial variable only. Further, q_t and q_w represent volumetric fluid sources and sinks (e.g., injection and production wells). Finally, $\lambda_t = \lambda_w + \lambda_o$ denotes the total mobility, where the mobility of each phase λ_j is given as the relative permeability k_{rj} of phase j divided by the phase viscosity μ_j ($j = o, w$) and $f_w = \lambda_w/\lambda_t$ is the fractional-flow function of water.

Streamline solvers are based on a sequential time-stepping procedure. First the known initial saturation distribution is used to compute the mobilities $\lambda_t(S)$ in **Eq. 1**, after which the pressure equation can be solved to give total velocity $\vec{u}$ and pressure distribution p. Next, the total velocity $\vec{u}$ is kept fixed in **Eq. 2**, while the saturation is advanced a given time step. The new saturation values are used to update the mobilities in **Eq. 1**, the pressure equation is solved again, and so on.

2.1. Multiscale Pressure Solver. Our multiscale method is based on a mixed finite-element formulation of the flow equation **Eq. 1** in which one computes an approximation to the pressure and velocity simultaneously. That is, one seeks a pair $(\vec{u}, p)$ in $\mathcal{U} \times \mathcal{V}$, such that

$$\int_\Omega \vec{u} \cdot (\lambda_t k)^{-1}\vec{w}\, dx - \int_\Omega p\nabla \cdot \vec{w}\, dx = 0, \qquad \forall \vec{w} \in \mathcal{U}, \tag{3}$$

$$\int_\Omega l\nabla \cdot \vec{u}\, dx = \int_\Omega ql\, dx, \qquad \forall l \in \mathcal{V}. \tag{4}$$

In a standard discretization, the finite-dimensional function spaces $\mathcal{U}$ and $\mathcal{V}$ for velocity and pressure, respectively consist of low-order piecewise polynomials. In the multiscale mixed finite-element method (MsMFEM) (*Chen and Hou*, 2002; *Aarnes*, 2004), the approximation space $\mathcal{U}_{\text{ms}}$ for velocity has a multiscale structure, whereas $\mathcal{V}$ is chosen simply as the space of of piecewise constant functions, since the pressure is immaterial for the incompressible flows

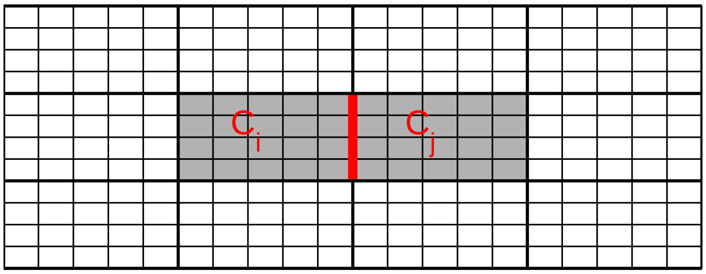

FIGURE 1. A coarse grid overlying a fine grid with the gray area giving support of basis function $\vec{\Psi}_{ij}$, which is associated with the edge/face indicated by the thick.

considered herein. If the pressure solution is needed, a more accurate pressure distribution can be constructed either by adaptively gridding around fluid sources and sinks, or by locally extracting a subgrid pressure distribution from the multiscale approximation space.

Basis Functions. Although MsMFEM can be defined for general unstructured grids (*Aarnes et al.*, 2006, to appear), we only consider Cartesian grids herein. Let $\{K_m\}$ be a (uniform) partitioning of Ω into mutually disjoint grid cells. Furthermore, let $\{C_i\}$ be a coarse partitioning of Ω, defined in such a way that each fine cell K_m overlaps with a single coarse block C_i, see **Fig. 1**. The multiscale approximation space $\mathcal{U}_{\mathrm{ms}}$ is defined by assigning a $\vec{\Psi}_{ij}$ to each non-degenerate interface between two coarse blocks, $\Gamma_{ij} = \partial C_i \cap \partial C_j$. The basis functions $\vec{\Psi}_{ij}$ are computed numerically by forcing unit flow from block C_i to C_j; that is, by solving a local flow problem in each pair of blocks $\Omega_{ij} = C_i \cup C_j$

$$\vec{\Psi}_{ij} = -\lambda_t k \nabla \Phi_{ij}, \quad \nabla \cdot \vec{\Psi}_{ij} = \begin{cases} w_i(x), & x \in C_i, \\ -w_j(x), & x \in C_j, \end{cases} \tag{5}$$

with $\vec{\Psi} \cdot \vec{n} = 0$ on the boundary of Ω_{ij}. To solve (5) we can use any consistent and mass-conservative method; here we use the standard two-point flux-approximation (TPFA) scheme. By choosing $w_i \propto q$ for coarse blocks containing sources or sinks, we ensure a conservative approximation to $\vec{u}$ on the fine grid. In all other blocks, we set $w_i \propto \mathrm{trace}(k(x))$. Moreover, to give a unit flow from C_i to C_j, the source terms $w_i(x)$ are normalized such that $\int_{C_i} w_i(x)\,dx = 1$; this is discussed in more detail by *Aarnes* (2004) and *Aarnes et al.* (2006). The corresponding basis functions can be seen as generalizations of the lowest-order Raviart–Thomas basis functions in a standard mixed method. **Figure 2** illustrates the x-component of the basis function for a homogeneous and a heterogeneous medium.

Selective Updating of Basis Functions. Solving local flow problems is typically the most expensive step in a multiscale method, and the overall computational cost of generating basis functions. Computing basis functions and solving the coarse-grid system is comparable to solving the pressure equation directly on the fine grid using a highly efficient linear solver, like e.g., algebraic multigrid (*Stüben*, 2000); a more detailed discussion is given by *Kippe et al.* (to appear). Huge computational savings can be obtained if basis functions can be computed only initially or recomputed infrequently throughout the simulation. From **Eq. 5**, we see that $\vec{\Psi}_{ij}$ depends on three quantities that may change from one pressure solve to the next: the total mobility λ_t, the absolute permeability k, and the forcing terms $w(x)$, which again are determined by k and q.

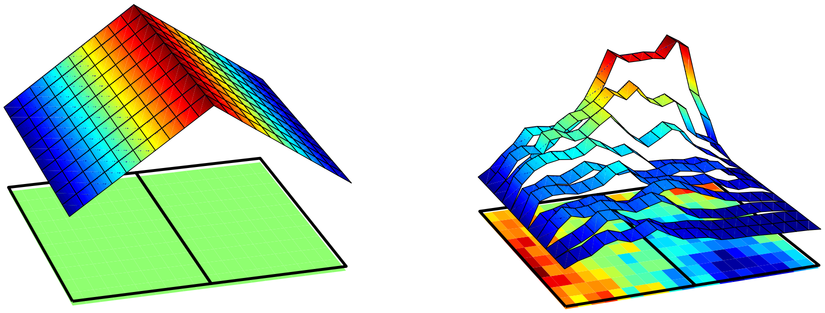

FIGURE 2. The x-component of the velocity basis function associated with an edge/face between two coarse blocks for a homogeneous and a heterogeneous permeability field, respectively.

Changes in fluid sources q may have a strong impact both on the local and global flow patterns, in particular if the changes are due to large changes in well rates, shut-in of wells, infill drilling, etc. Basis functions containing a well within their support should therefore be recomputed whenever the well configuration changes significantly.

The absolute permeability k may in principle change from one forward simulation to the next. Changes in k are accounted for on the coarse scale (**Eq. 3**), but not in the local basis functions (**Eq. 5**) unless these are updated. However, using a 'wrong' basis function may not have a significant impact on the calculated dynamic data. As a simple means for detecting changes in the permeability that significantly affect calculated dynamic data, we suggest to use the production-response sensitivities to be introduced in Section 4. A single sensitivity coefficient can be assigned to each coarse block by summing the sensitivities over the underlying fine grid; see *Yoon et al.* (2001). Due to the low resolution of dynamic data and the use of spatial regularization terms in the inversion process, changes in absolute permeability k from one inversion step to the next will mainly appear in regions of high sensitivity. Basis functions should therefore be recomputed initially in regions with high sensitivity; in the rest of the reservoir, reasonable accuracy is obtained by reusing basis functions from the previous forward simulation.

Changes in total relative mobility $\lambda_t(S)$ are relatively smooth, unless a strong saturation front passes through the block, and can be accounted for on the coarse scale (**Eq. 3**) with reasonable accuracy. Moreover, changes in total mobility in high-sensitivity regions will have a stronger influence on the dynamic reservoir responses, indicating that basis functions in these regions should be updated dynamically throughout the simulation. The errors induced by not updating basis functions in low-sensitivity regions will have a limited effect on the dynamical reservoir responses, and one may therefore avoid updates there.

Our selective updating strategy is summarized as follows (*Stenerud et al.*, to appear): In the first forward simulation of the inversion procedure, we typically update all basis functions in every pressure step, because no sensitivities are yet available. After the first forward simulation, we sort the sensitivity coefficients of the coarse blocks in ascending order and mark a predefined fraction to be updated. We will refer to the strategy where $x\%$ of the basis functions are updated initially and the remaining $(100-x)\%$ are kept from the previous flow simulation, as $x\%$ initial update. Similarly, the coarse-grid sensitivities are used to pick blocks in which we may avoid dynamic updates of basis functions from one pressure step to the next. We refer to this as $x\%$ dynamical update when $x\%$ of the basis functions are updated dynamically each time step. We will assess the efficiency and robustness of this strategy in Section 4.

2.2. Streamline Solver. Instead of discretizing and solving the transport directly on the given grid, a streamline method decouples the 3-D equation, **Eq. 2**, into multiple 1-D equations along streamlines. To parameterize the streamlines, we introduce the time-of-flight variable τ defined by,

$$\tau(r) = \int_0^r \frac{\phi(\zeta)}{|\vec{u}(\zeta)|}\, d\zeta = \int_0^r s(\zeta)\, d\zeta, \tag{6}$$

which expresses the time it takes a passive particle to travel a distance r along a streamline. In differential form **Eq. 6** reads $\vec{u} \cdot \nabla\tau = \phi$. Using the bi-streamfunctions ψ and χ (*Bear*, 1972), for which $\vec{u} = \nabla\psi \times \nabla\chi$, we can define an alternative 3-D curvilinear coordinate system $(\tau,\ \psi,\ \chi)$, where the velocity $\vec{u}$ and hence the streamlines are orthogonal to the ψ and χ axes. In the streamline coordinates $(\tau,\ \psi,\ \chi)$, the gradient operator is expressed as

$$\nabla_{(\tau,\psi,\chi)} = (\nabla\tau)\frac{\partial}{\partial\tau} + (\nabla\psi)\frac{\partial}{\partial\psi} + (\nabla\chi)\frac{\partial}{\partial\chi}. \tag{7}$$

Because $\vec{u}$ is orthogonal to $\nabla\psi$ and $\nabla\chi$, it follows that $\vec{u} \cdot \nabla = \phi\frac{\partial}{\partial\tau}$, which together with the incompressibility condition $\nabla \cdot \vec{u} = 0$ can be used to rewrite **Eq. 2** as a family of one-dimensional transport equations along streamlines

$$\frac{\partial S}{\partial t} + \frac{\partial f_w}{\partial\tau} = 0. \tag{8}$$

The solution of **Eq. 2** is obtained by tracing numerous streamlines, mapping the initial saturations from the 3-D pressure grid to 1-D streamlines, and then solving **Eq. 8** along each streamline. Afterwards, the new streamline saturations are mapped (or averaged) back to the underlying 3-D grid to update mobilities before the pressure equation is solved to recompute the velocity field.

To trace streamlines, most streamline solvers use a simple semi-analytical procedure due to *Pollock* (1988), by which each streamline is traced numerically cell-by-cell, either from injector to producer, or vice versa, or from an arbitrary point in the reservoir and forward to fluid sinks and backward to fluid sources. After the tracing, each streamline ℓ is given as the indices of the cells the streamline traverses, the entry and exit points, and the incremental time-of-flights $\{\Delta\tau_{\ell,i}\}$ for each cell i. These increments form the cells in the streamline grid, on which **Eq. 8** will be solved. Initial values for **Eq. 8** are obtained by picking up the piecewise constant values from the underlying (pressure) grid,

$$S_{\mathrm{sl},i} = S_i. \tag{9}$$

To solve **Eq. 8**, we will herein use a front-tracking method (*Holden and Risebro*, 2002) that is unconditionally stable and can directly utilize the time-of-flight grid resulting from the streamline trace. This makes the method very efficient and devoid of numerical diffusion. In contrast, solvers based on a finite-volume formulation typically need to map the initial data to a more regular grid.

To map values from the streamlines back to the underlying 3-D grid, we use volumetric averaging. Volumes are associated with streamlines by considering each streamline as a representation of the cross-section of a streamtube with an associated constant volumetric flux $q_\ell = |\vec{u}(\zeta)|A(\zeta)$. This gives the volume of the streamline as

$$(10)\qquad \begin{aligned} V_\ell &= \int_0^s \phi(\zeta)A(\zeta)\,d\zeta \\ &= q_\ell \int_0^s \frac{\phi(\zeta)}{|\vec{u}(\zeta)|}\,d\zeta = q_\ell\,\tau_\ell. \end{aligned}$$

The volume of a streamline in grid cell i is then $V_{\ell,i} = q_\ell \Delta\tau_{\ell,i}$, and the precise definition of the streamline-to-grid volumetric averaging is,

$$(11)\qquad S_i = \frac{\sum_\ell S_{\ell,i}V_{\ell,i}}{\sum_\ell V_{\ell,i}}.$$

We note that considering streamlines as fluid carriers also makes it natural to define production characteristics simply by summing the outflow fluxes during time step Δt from all streamlines connected to each well

$$(12)\qquad \mathrm{PRD}_{\Delta t} = \sum_\ell q_\ell \int_{\Delta t} f_{w,\ell}(t)\,dt.$$

To associate fluxes to each streamline, we generate equally spaced starting points on the faces of grid cells containing injection wells. The number of starting points on each face is proportional to the volumetric flux across the face; i.e., streamlines carry approximately equal amounts of fluids, $q_\ell \approx C$. An advantage of this approach is that the sums in **Eqs. 11** and **12** can be computed incrementally as streamlines are traced (*Batycky*, 1997) without knowing the associated volumetric flux, thus allowing completely independent processing of streamlines.

For the volumetric mapping **Eq. 11** to make sense, each grid cell should in principle be traversed by at least one streamline. In general, there will be a number of grid cells that are not traversed by any of the streamlines traced from the faces of injector-cells. One can therefore perform an additional tracing process, where one picks a point inside one of the untraced cells and traces a streamline from this point and backward/forward to a fluid source/sink or to a cell that has been traversed by another streamline (*Batycky*, 1997). This is repeated until there are no untraced cells. Alternatively, one may simply ignore the untraced cells, as these often are in regions that contribute little to the production characteristics. To keep the amount of streamline tracing at a minimum, we here employ the latter approach.

3. Improving Local and Global Mass Balance

Lack of mass conservation is a well-known problem for streamline simulators and may lead to both incorrect saturation distributions and incorrect production curves. To illustrate typical errors observed as the number of streamlines is reduced, we consider a large 3-D reservoir model of a Brent sequence consisting of 60 × 220 × 85 grid cells, see (*Christie and Blunt*, 2001) for more details. The reservoir is produced using a five-spot pattern of vertical wells; the central injector has a constant rate of 5 000 bbl/day (reservoir conditions), and the four producers operate at 4 000 psi bottom-hole pressure. We assume quadratic relative permeability curves with $S_{wc} = S_{or} = 0.2$. The initial saturation is $S_0 \equiv S_{wc}$, and the viscosities are $\mu_o = 3.0$ cP and $\mu_w = 0.3$ cP, respectively. We neglect gravity and compressibility, since these have smaller impact on the production curves than the numerical diffusion inherent in any numerical scheme. Moreover, for the pressure equation we use a

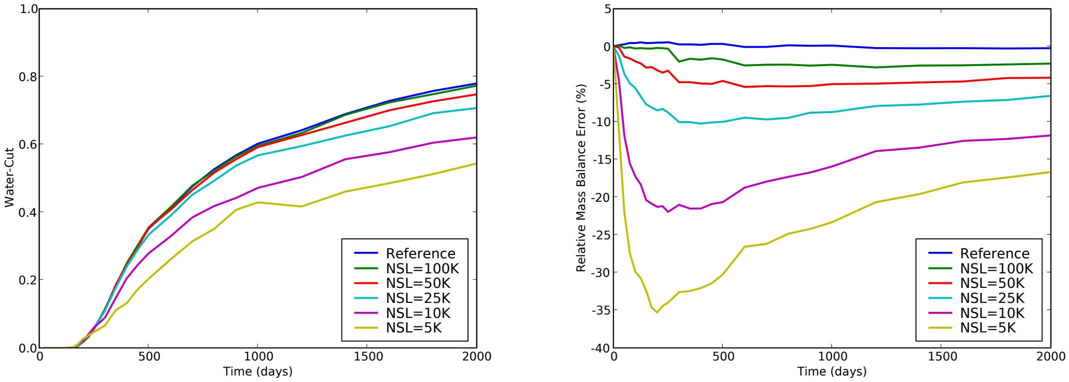

FIGURE 3. Fractional flow (left) and relative mass-balance errors (right) for Producer 1 from Model 2, SPE 10 for various number of streamlines (1K = 1 000).

standard two-point discretization with an AMG linear solver (*Stüben*, 2000). The time-steps are those reported for the commercial streamline simulator used in the study.

Figure 3 shows fractional-flow curves in Producer 1 for simulations with various number of streamlines. The water production is clearly underestimated when the number of streamlines is too small. Since the correct total amount of injected water is distributed among streamlines at the injecting end of each streamline, there must effectively be a loss of mass in the method. We can quantify this loss by the relative *global mass-balance error* for water in each time-step,

$$\epsilon_{\Delta t} = \frac{\mathrm{INJ}_{\Delta t} - \mathrm{PRD}_{\Delta t} + \mathrm{FIP}_t - \mathrm{FIP}_{t+\Delta t}}{\mathrm{INJ}_{\Delta t}}, \tag{13}$$

which is equivalent to an error in the volume balance, since we have assumed incompressibility. **Figure 3** also shows that the errors increase rapidly in the beginning of the simulation and decay slowly as the fractional-flow curves increase. Notice that since production curves are calculated directly from the individual streamlines using **Eq. 12**, inaccurate production curves do not necessarily imply inaccurate saturation distributions, and vice versa.

3.1. Global Mass-Balance Errors. Viewing streamlines as fluid carriers introduces a fundamental problem in that the pore volume represented by a finite number of streamlines does not necessarily match the pore volume of the original grid (in physical space); in other words, the two grids are not automatically compatible. This will generally lead to mass-balance errors when mapping saturation between the streamlines and the pressure grid. From **Eq. 10** we have that the streamline pore volume is given by,

$$\mathrm{V}_{\mathrm{sl}} = \sum_{\ell} q_\ell \, \tau_\ell. \tag{14}$$

Thus, the flux q_ℓ and the total time-of-flight τ_ℓ associated with streamline ℓ are two parameters we can play with to improve the mass-balance properties of our streamline discretization. Both parameters are generally subject to approximation errors.

Using the semi-analytical streamline tracing method introduces errors in τ_ℓ, even for Cartesian grids with given analytical fluxes on the faces, see (*Matringe and Gerritsen*, 2004).

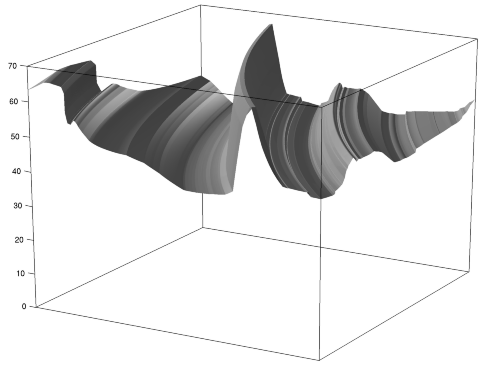

FIGURE 4. Time-of-flight in grid cell $(200, 36)$ of Layer 76 in Model 2 of SPE 10 sampled in 2000×2000 evenly distributed points inside each cell. Here the variation of τ is of the same order as the values of τ.

However, for the Cartesian geometries considered herein, careful numerical studies revealed that Pollock's method was sufficiently accurate.

Errors in the fluxes q_ℓ correspond to errors in the *transversal* discretization in (ψ, χ), which is determined implicitly by the distribution of streamlines and association of fluxes. The fluxes q_ℓ represent velocity integrals over the cross-section of the associated streamtubes. Assigning equal fluxes $q_\ell = C$ to all streamlines at the injector may be inaccurate and will in particular mean that the total flux of all streamlines passing through a particular cell face elsewhere in the reservoir will not necessarily match the corresponding flux in the pressure grid. However, only minor improvements in the mass conservation were observed when using more accurate assignment of fluxes, e.g., by scaling q_ℓ according to the interpolated velocity at the starting point and the cross-section area of the associated streamtube as discussed by *Ponting* (1998) and *Pallister and Ponting* (2000). Lifting the restriction of equal streamline fluxes also makes it possible to apply other streamline distribution schemes. For instance, in situations where there is a large variation in total fluid rates between different fluid sinks (producers), it may be beneficial to start streamlines also on the faces of cells containing sinks to ensure that sufficient accuracy is achieved for sinks with small rates. Similarly, streamline fluxes may be assigned at the faces of producers or as a weighted averages of the flux at the injector and the producer. None of these ideas had a significant effect for the applications considered herein, see (*Kippe et al.*, 2007) for more details.

In our experience, the *global* mass-balance errors we observe as the number of streamlines is reduced are primarily caused by the fact that τ_ℓ may not be a good approximation to the average time-of-flight over cross-sections of the associated streamtube. This is illustrated in **Fig. 4**, which shows the time-of-flight sampled at 2000×2000 evenly distributed points within a single cell of Layer 76 in the SPE 10 data set. Here the variation of τ is of the same order as the values of τ itself. Increasing the number of streamlines decreases the streamtube cross-sections and hence reduces this error. However, considering the very large variation in τ shown in **Fig. 4**, it is evident that a large number of streamlines is necessary to obtain accurate streamline volumes and thereby low error in the global mass balance. On the other hand, if we insist on keeping the number of streamlines low, we can use the fact that mass should be conserved, and correct the computed values of τ_ℓ to enforce the mass-balance constraint.

3.2. Improved Accuracy of Production Curves. Exact global conservation of mass is guaranteed if the streamline volume matches the true pore volume, i.e., $\sum_\ell V_{\ell,i} = V_i$, in every grid-block touched by streamlines. In this case, the mappings back-and-forth between streamlines and the pressure grid preserve mass. Indeed, for the mapping from grid to streamlines, **Eq. 9**, we have,

$$V^w_{\text{grid}} = \sum_i V_i S_i = \sum_i \Big(\sum_\ell V_{\ell,i}\Big) S_i = \sum_\ell \sum_i V_{\ell,i} S_{\ell,i} = V^w_{\text{sl}}, \tag{15}$$

and similarly for the mapping from streamlines to grid, **Eq. 11**,

$$V^w_{\text{sl}} = \sum_\ell \sum_i V_{\ell,i} S_{\ell,i} = \sum_i \sum_\ell \frac{V_i}{\sum_\ell V_{\ell,i}} V_{\ell,i} S_{\ell,i} = \sum_i V_i S_i = V^w_{\text{grid}}. \tag{16}$$

Here V^w_{grid} and V^w_{sl} are the total volumes of water on the pressure and streamline grids, respectively. Since the streamline flux is constant along each streamline, our only option for ensuring $\sum_\ell V_{\ell,i} = V_i$ is to modify the local time-of-flight increments, $\Delta\tau_{\ell,i}$. Specifically, prior to solving the one-dimensional saturation equation **Eq. 8** along streamlines, we propose to scale the time-of-flight values $\tau_{i,\ell}$ in block i by a factor $\alpha_i = V_i / \sum_\ell V_{\ell,i}$. This means that streamlines can no longer be processed independently, and we need to store streamlines in memory, or alternatively perform the complete tracing procedure twice; once to compute the values of α_i, and then a second time for the solution of the one-dimensional problems. The memory required to store streamlines is usually (significantly) less than the memory required to solve the pressure equation **Eq. 1**. Hence, we prefer storing rather than retracing, since tracing is an expensive process.

Scaling the time-of-flight amounts to locally stretching or shrinking the grid on which **Eq. 8** is solved. By enforcing mass conservation we thus introduce local errors in the saturation distribution, but as we demonstrate below, the global properties of the resulting solutions are better. However, special care must be taken to not ruin important (local) characteristics like the breakthrough-time for producers, which e.g., will be important in the inversion procedure discussed below. To make sure breakthrough is estimated correctly, we only apply the scaling along streamlines after breakthrough has occurred.

In **Fig. 5** we have recomputed the simulation reported in **Fig. 3**, but now correcting for incorrect streamline volumes. The mass-balance errors are still large initially since the time-of-flight scaling is only applied after breakthrough, but the errors decrease rapidly. The improvement of the fractional-flow curves is significant, to say the least, with as few as 5 000 streamlines giving acceptable results. **Table 1** reports the errors in the fractional-flow curves $w(t)$ for all four producers, as measured by

$$\delta(w) = \|w - w^{\text{ref}}\|_2 / \|w^{\text{ref}}\|_2. \tag{17}$$

For completeness, **Table 1** also shows the corresponding results for the standard streamline approach, where we have started streamlines in both injectors and producers and used weights given by the area of the perpendicular bisection of the cell faces to assign fluxes to streamlines (since this gives slightly better results for the original method). Moreover, the table reports

Table 1. Errors in fractional flow $\delta(w)$ for producers P1 to P4 and average saturation error $\delta(S)$ for the original (O) and modified (M) streamline methods on Model 2, SPE 10 for various number of streamlines (NSL). Columns T_{sl} and T_{tot} report the total computational time for the streamline solves and the overall simulation, respectively, measured on a workstation PC with a 2.4 GHz Intel Core 2 Duo processor with 4 Mb cache and 3 Gb memory.

NSL	O/M	P1	P2	P3	P4	$\delta(S)$	T_{sl} (s)	T_{tot} (s)
100 000	O	8.91e-03	6.24e-03	2.44e-03	2.99e-03	2.75e-02	508.92	974.94
	M	9.86e-03	4.61e-03	1.97e-03	3.67e-03	2.83e-02	508.20	979.03
50 000	O	2.53e-02	1.72e-02	6.42e-03	9.38e-03	4.00e-02	266.48	728.42
	M	1.66e-02	7.88e-03	3.72e-03	7.03e-03	3.81e-02	265.87	727.79
25 000	O	6.49e-02	4.85e-02	1.74e-02	2.28e-02	5.89e-02	147.36	608.46
	M	1.43e-02	1.47e-02	8.12e-03	7.12e-03	5.27e-02	146.23	613.00
10 000	O	1.78e-01	1.29e-01	5.53e-02	7.30e-02	9.54e-02	75.65	541.17
	M	3.26e-02	1.94e-02	1.56e-02	1.38e-02	8.06e-02	75.33	545.09
5 000	O	3.20e-01	2.30e-01	1.02e-01	1.30e-01	1.29e-01	50.91	512.75
	M	4.25e-02	2.19e-02	1.86e-02	2.37e-02	1.12e-01	51.74	516.63

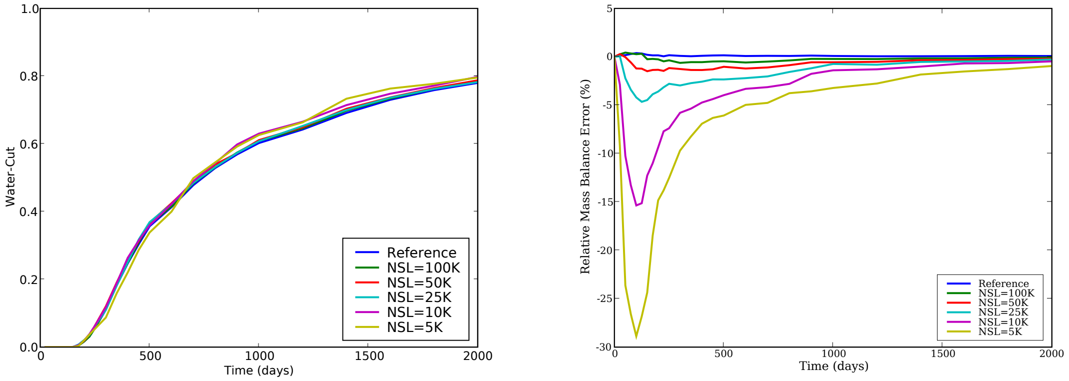

Figure 5. Fractional flow (left) and relative mass-balance errors (right) for Producer 1 from Model 2, SPE 10 for various number of streamlines when using the modified streamline method (1K = 1 000).

saturation errors in the the porosity-weighted L^1-norm,

$$\delta(S) = \|\phi(S - S^{\text{ref}})\|_1 / \|\phi S^{\text{ref}}\|_1, \tag{18}$$

averaged over all time steps of the simulation. Altogether, these results show that although scaling the time-of-flight values has limited effect on the accuracy of the saturation fields, the accuracy of the corresponding production curves is improved significantly. For instance, if one is primarily interested in the fractional-flow curves and allows an error of about 5%, it is sufficient to use only 5 000 streamlines for the modified method, whereas 25–50 000 streamlines would be required in the original method. This yields a significant speedup for the transport part of the simulation, since the computation time associated with transport in theory scales linearly with the number of streamlines. The timing results in **Table 1** show that the actual

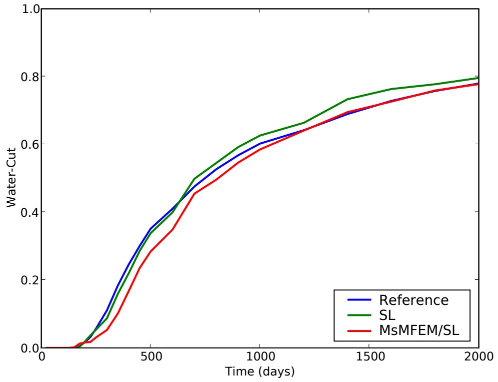

FIGURE 6. Fractional flow in Producer 1 for the modified streamline with 5 000 streamlines, using a standard two-point and the MsMFEM pressure solver.

scaling is not truly linear as the number of streamlines becomes very small. However, this is to be expected since our simulator is optimized for relatively large numbers of streamlines and otherwise negligible overhead associated with streamline distribution, flux computations, and saturation mappings may become significant when using a small number of streamlines. Still, we see that going from 50 000 to 5 000 streamlines gives at least five times speedup for the transport step.

As the number of streamlines is reduced, the total simulation time in **Table 1** is dominated by the solution of the pressure equation, **Eq. 1**. To obtain a more substantial speedup for the overall simulation, we use the multiscale pressure solver introduced in Section 2.1 on a 5 × 11 × 17 coarse grid. The fractional-flow curves shown in **Fig. 6** demonstrate that utilizing MsMFEM for the pressure equation does not yield a significantly reduced accuracy in the production curves for the case with 5 000 streamlines, but the overall simulation time is reduced from 8 minutes and 36 seconds to an impressive 2 minutes and 22 seconds.

Finally, we emphasize that scaling the time-of-flight is primarily aimed at improving the *global mass balance* by increasing the accuracy of measuring reservoir production (i.e., global, low-resolution flow responses). In the next section we will look at another technique more aimed at improving local, high-resolution flow responses (pointwise saturation distributions).

3.3. Improving Local Mass Balance – Adaptive Streamline Coverage. The correction strategy introduced above is no guarantee for producing accurate saturation curves or fractional-flow curves. Indeed, insufficient streamline coverage may still induce large errors (for piston-like displacements) if we do not ensure that all grid cells are traversed by streamlines. This leads to errors in the computed pressure and velocity fields, thus shifting the predicted time of breakthrough. For scenarios with high mobility ratios, the pressure/velocity solutions are less sensitive to errors in the underlying saturation field, because the saturation variation is generally much smoother. On the other hand, our correction strategy never performs significantly worse than the original method, and can therefore always be applied safely.

To alleviate the accuracy problems for favorable displacement conditions, we could trace streamlines through every cell, using, e.g., the approach of *Batycky* (1997). However, many

Table 2. Errors in fractional flows ($\delta(w)$) and average number of streamlines (NSL) on the homogeneous model for the original and the adaptive streamline tracing for end-point mobility ratio $M_{\text{end}} = 0.1$.

β	NSL	P1	P2	P3	P4
—	2000	4.43e-02	9.30e-02	8.97e-02	9.47e-02
—	1500	5.27e-02	1.13e-01	8.96e-02	1.04e-01
—	1000	1.39e-01	1.62e-01	1.76e-01	1.67e-01
—	500	4.25e-01	4.34e-01	4.64e-01	4.61e-01
1.0	873	1.17e-02	7.52e-03	2.44e-02	1.37e-02
0.9	701	3.28e-02	2.95e-02	4.82e-02	2.17e-02
0.8	560	2.40e-01	2.31e-01	2.72e-01	2.43e-01
0.7	500	3.34e-01	3.85e-01	3.99e-01	3.90e-01
0.6	500	3.60e-01	3.78e-01	3.98e-01	3.86e-01

Table 3. Errors in fractional flow ($\delta(w)$) and average number of streamlines (NSL) on the homogeneous model for the original and the adaptive streamline tracing for end-point mobility ratio $M_{\text{end}} = 10$.

β	NSL	P1	P2	P3	P4
—	2000	2.58e-02	2.45e-02	2.33e-02	8.44e-03
—	1500	3.14e-02	1.00e-02	3.88e-02	9.23e-03
—	1000	6.68e-02	2.29e-02	5.79e-02	4.14e-02
—	500	7.42e-02	9.58e-02	1.20e-01	8.96e-02
1.0	873	3.45e-02	2.21e-02	2.26e-02	2.04e-02
0.9	722	3.42e-02	2.39e-02	2.41e-02	2.79e-02
0.8	616	2.69e-02	2.44e-02	3.35e-02	2.77e-02
0.7	519	2.19e-02	2.50e-02	5.94e-02	2.43e-02
0.6	500	2.39e-02	3.60e-02	6.80e-02	3.49e-02

cells will typically be located in low-flow regions that do not significantly affect the solution. We therefore propose an adaptive approach to streamline coverage, where we only demand that a given fraction β of the pore volume should be traversed by streamlines. Before the tracing starts, the cells are sorted in descending order by absolute velocity $|\vec{u}|$, and we trace back from untouched blocks in sorted order until the given pore-volume target has been met. We also ensure that each well is properly covered by starting a specified number of streamlines from fluid sources/sinks, with the distribution of streamlines on the faces of grid cells containing fluid sources/sinks given according to the fluxes, as before.

To demonstrate the effect of the adaptive tracing, we consider a homogeneous $32 \times 32 \times 8$ model of aspect ratio $1 : 1 : 0.1$, with wells placed in a five-spot pattern, where the four producers operate at equal bottom-hole pressures. We assume quadratic relative permeability curves with zero residual oil and water saturations, and perform simulations for two different values of the end-point mobility ratio $M_{\text{end}} = \mu_o/\mu_w$; favorable displacement ($M_{\text{end}} = 0.1$) and unfavorable displacement ($M_{\text{end}} = 10$). The dimensionless simulation time is 2.0 PVI, and for both displacement scenarios we verified that the chosen number of time-steps was sufficient for stability of the sequential time-stepping scheme.

Tables 2 and **3** show the average number of streamlines and errors in the fractional-flow curves when applying the adaptive approach in combination with the modified streamline method. Initially we trace 100 streamlines from each well, which is why the minimum number of streamlines is 500. Compared with the non-adaptive version, the adaptive method gives significantly more accurate production curves using fewer streamlines. As expected, the optimal value of β depends on the displacement conditions, with favorable piston-like displacement requiring a larger fraction of the pore volume to be covered. In the unfavorable case, the production curves are good even without the adaptivity. In fact, we could actually have used even fewer streamlines. This helps explain why we obtained accurate results using *very* few streamlines for the SPE 10 model above.

4. Integration of Dynamic Data

In this section we present the inversion method in more detail and discuss its efficiency and robustness using three numerical test cases. In the following we assume that the subsurface model has been conditioned to static data and available geological information, such that the model already gives a reasonable description of the reservoir geology. Dynamic production data generally have low resolution and cannot be used to infer small-scale variations in reservoir properties. It is therefore important that changes to the subsurface model inferred from the dynamic data are kept as minimal as possible to preserve geological realism.

4.1. Generalized Travel-Time Inversion. Our method for integrating fractional-flow data utilizes approximate sensitivities calculated analytically along streamlines to update the heterogeneous subsurface model based on observed dynamic data (*Vasco et al.*, 1999; *He et al.*, 2002). The sensitivities quantify the influence of reservoir parameters on dynamical responses of the reservoir. As such, these sensitivities provide the fundamental relationships that allow us to integrate the dynamic reservoir responses. The major steps in our method are:

(1) Multiscale-streamline simulation to compute production responses at the observation points (wells) as discussed in Section 2.
(2) Quantification of the mismatch between observed and computed dynamic responses via a generalized travel-time formulation. An optimal travel-time shift is computed for each observation point (e.g., production well) by systematically shifting the computed production responses towards the observed data until the cross-correlation between the two is maximized (*He et al.*, 2002).
(3) Computation of streamline-based analytic sensitivities of the production responses (fractional-flow curves) to reservoir parameters, specifically permeability.
(4) Updating of reservoir properties to match the dynamical reservoir responses (production data) via inverse modeling. To this end, we will use an iterative least-square minimization algorithm (LSQR) (*Vasco et al.*, 1999; *He et al.*, 2002) to simultaneously minimize the travel-time misfit for all observation points, thereby matching all dynamic data within a specified tolerance.

This four-step process is repeated until a satisfactory match is obtained. Next, we describe the three last steps in more detail, starting with the formulation of a generalized travel-time misfit to quantify mismatch in dynamic data.

Misfit in dynamic data is commonly represented by a least-squares functional of the form:

$$E = \sum_k \sum_j \Big[w_k^{\mathrm{obs}}(t_j^k) - w_k^{\mathrm{calc}}(t_j^k) \Big]^2 , \tag{19}$$

where $w_k^{\text{obs}}(t_j^k)$ and $w_k^{\text{calc}}(t_j^k)$ are the observed and calculated data, respectively, in well k at time t_j^k. Direct minimization of **Eq. 19** is called amplitude inversion, in which the observation times are fixed and one seeks to match the amplitudes. Travel-time inversion, on the other hand, chooses a specific point on the dynamic data curve (e.g., the breakthrough time or a distinct peak) and adjusts the model parameters so that a similar point is obtained in the computed reservoir response. Although crude, this approach has an important advantage: whereas amplitude inversion is highly nonlinear, travel-time inversion has quasilinear properties (*Cheng et al.*, 2005) and is thus more robust and less likely to be stuck in local minima. However, the resulting overall data match of dynamic data may not be satisfactory since only a single data point is matched (per well).

The generalized travel-time inversion combines the desirable properties of travel-time and amplitude inversion into one step (*He et al.*, 2002) by seeking a set of optimal time-shifts $\mathbf{\Delta t} = \{\Delta t_k\}$ that minimize the following misfit at each well:

$$E_k(\Delta t_k) = \sum_j \left[w_k^{\text{obs}}(t_j^k + \Delta t_k) - w_k^{\text{calc}}(t_j^k) \right]^2 . \tag{20}$$

Hence we can match multiple data points as in the amplitude inversion, while retaining the attractive quasilinear properties of the travel-time inversion. Computing time-shifts does not require any new flow simulation, but can be done using data from the single forward simulation used to evaluate the data mismatch.

Having determined the optimal time-shifts, the next step is to propagate them into changes in the reservoir parameters. Mathematically, the inversion of the time-shifts $\mathbf{\Delta t}$ leads to the minimization of a penalized misfit function (*Vasco et al.*, 1999; *He et al.*, 2002):

$$\|\mathbf{\Delta t} - \mathbf{G}\delta\mathbf{m}\| + \beta_1\|\delta\mathbf{m}\| + \beta_2\|\mathbf{L}\,\delta\mathbf{m}\|. \tag{21}$$

Here $\delta\mathbf{m}$ denotes the changes in the reservoir properties $\mathbf{m}$, $\mathbf{G}$ contains the sensitivities of the time shifts with respect to the reservoir parameters $\mathbf{m}$, and $\mathbf{L}$ is a second-order (Laplace) difference operator. The first term ensures that the difference between the observed and calculated dynamic responses is minimized. The second term is a norm constraint that penalizes deviations from the initial (prior) subsurface model and as such helps to preserve the geological realism of the inversion. The third term is a roughness constraint that measures the regularity of the changes and is introduced to stabilize the inversion by only allowing large-scale changes that are consistent with the low resolution of the production data. The weights β_1 and β_2 determine the relative strengths of the two regularization terms. The minimum in **Eq. 21** can be obtained by the iterative least-square minimization algorithm, LSQR (*Paige and Saunders*, 1982), for which the computational cost scales linearly with respect to the number of degrees-of-freedom (*Vega et al.*, 2004).

4.2. Time-Shift Sensitivities. For the sake of completeness, we briefly describe the analytical calculation of streamline-based sensitivities, which can be computed using a single flow simulation, leading to very fast algorithms for data integration or inverse modeling. Because the sensitivities are simple integrals along streamlines, the computation time scales very favorable with respect to the number of grid cells, thus making streamlines the preferred approach for integrating dynamic data into highly-detailed subsurface models.

The sensitivity of the shift in travel time Δt_k with respect to reservoir parameter m is given by (*Vasco et al.*, 1999; *He et al.*, 2002)

$$\frac{\partial \Delta t_k}{\partial m} = -\frac{1}{N_d}\sum_{j=1}^{N_d} \frac{\partial t_j^a}{\partial m}, \tag{22}$$

where N_d represents the number of observed data for the associated well and t_j^a is the common (or average) arrival time at the well for the connected streamlines. The sensitivity of the common arrival time with respect to a reservoir parameter m, is calculated by a flux-weighted average of the arrival-time sensitivities of the connected streamlines. The arrival-time in each streamline is related to the streamline time-of-flight by assuming a Buckley–Leverett profile

$$\tau_\ell = t_{j,\ell}^a \cdot \tilde{f}_w'(S). \tag{23}$$

Here $\tilde{f}$ denotes the convex hull of the fractional flow curve, and the derivative $\tilde{f}_w'$ is evaluated using the saturation at the outlet of streamline ℓ for streamlines with breakthrough, and using the front saturation for streamlines without breakthrough. In other words, $\partial t_{j,\ell}^a/\partial m$ is proportional to the sensitivity of the time-of-flight, which can be computed analytically from a single streamline simulation under the assumption that the streamlines do not shift because of small perturbations in reservoir properties. For example, the sensitivity with respect to permeability k_i in cell K_i is given by

$$\frac{\partial \tau}{\partial k_i} = \int_{\Sigma_i} \frac{\partial s(\zeta)}{\partial k_i}\, d\zeta = -\int_{\Sigma_i} \frac{s(\zeta)}{k_i}\, d\zeta = -\frac{\Delta\tau_i}{k_i}, \tag{24}$$

where the integral is along the streamline trajectory Σ_i through K_i and $\Delta\tau_i$ is the associated incremental time-of-flight. Similarly, sensitivities can be calculated with respect to mobility or to the product of mobility and permeability. Worth mentioning here is an important practical aspect. Our experience indicates that the selective work-reduction strategy and the data-integration process are more robust if the sensitivities are made dimensionless as described by *He et al.* (2002).

We are now fully equipped to integrate dynamic data into high-resolution subsurface models. In (*Stenerud et al.*, to appear), the accuracy and robustness of our inversion method were investigated for a small 2-D case with isotropic lognormal permeability and flow with end-point mobility ratios ($M_{\text{end}} = 0.2, 0.5, 10$). In the next two subsections, we investigate the accuracy and robustness of the proposed selective updating of basis functions more thoroughly using two 2-D test cases that involve dynamic well configuration and multiple realizations, respectively. *To pose a further challenge for our multiscale simulator, both cases involve anisotropic permeability structures with long streaks of high permeability aligned* exactly *with the diagonal direction of the grid.* As noted by *Kippe et al.* (to appear), this particular permeability structure is a worst-case scenario for MsMFEM, where the solver may exhibit loss of accuracy. For all other cases, the solver is generally very robust and accurate (*Kippe et al.*, to appear).

To measure the quality of the data integration, we use the *amplitude residual* $\sqrt{E}$ (see **Eq. 19**) and the *time-shift residual* $\|\mathbf{\Delta t}\|_2$. We also report the average discrepancy between the reference and matched permeability field measured by

$$\overline{\Delta \ln k} = \frac{1}{N}\sum_{i=1}^{N} \left| \ln k_i^{\text{ref}} - \ln k_i^{\text{match}} \right|. \tag{25}$$

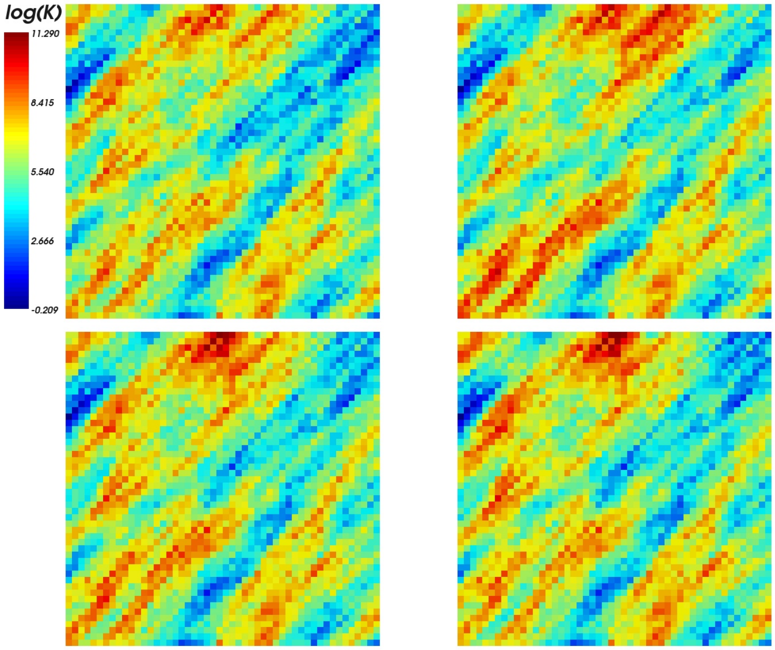

Figure 7. The plots show, from upper left to lower right, $\ln(k)$ for the reference permeability field, the initial permeability field, and the match obtained after eight iterations with and without updating basis functions.

4.3. Changing Well Conditions. We first consider a 2-D reservoir model with diagonal permeability streaks and a dynamic well configuration. The lognormal permeability field is given in terms of 50×50 uniform cells and has a diagonal structure with long correlation length, see **Fig. 7**. As above, we assume quadratic relative permeability curves with zero residual oil and water saturations and end-point mobility ratio $M_{\text{end}} = \mu_o/\mu_w = 0.5$. The forward simulator is run with pressure steps of 80 days, and for the MsMFEM pressure solver we construct a uniform 10×10 coarse where each block contains 5×5 fine cells.

Synthetic dynamic data were generated by adding 5% white noise to the fractional-flow curves computed from the reference permeability using a streamline simulator with a two-point pressure solver. Initially, the well configuration is a five-spot configuration, where the four producers operate with equal constant rate. The producer in the south-west corner has early breakthrough and is therefore converted to an injector after 640 days. Simultaneously, two new producers are introduced in the middle of each opposite boundary (north and east). After conversion and infill drilling, the south-west well is injecting 75% of the total injection rate and all producers are producing at equal constant rate. The motivation for the updated well configuration is to introduce an additional sweep from the south-west corner towards the opposite boundaries. The updated well configuration is kept throughout the rest of the production period. Hence, we wish to integrate 2000 days of production data from six producers in total.

To match observed data, we start from the prior permeability field shown in **Fig. 7** and treat the permeability in each cell as an adjustable parameter, giving a total of 2500 unknown parameters to be estimated. The time-shift sensitivities for each well are plotted in **Fig. 8**. The sensitivities are quite distinct and localized in channels due to the diagonal permeability

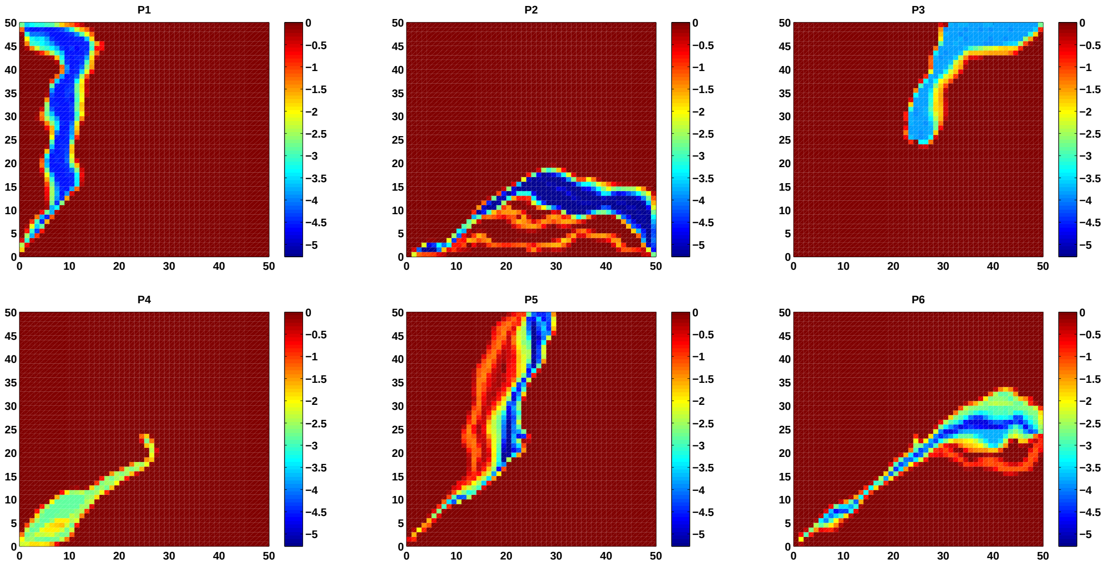

Figure 8. Streamline-based travel-time shift sensitivities for the six producers.

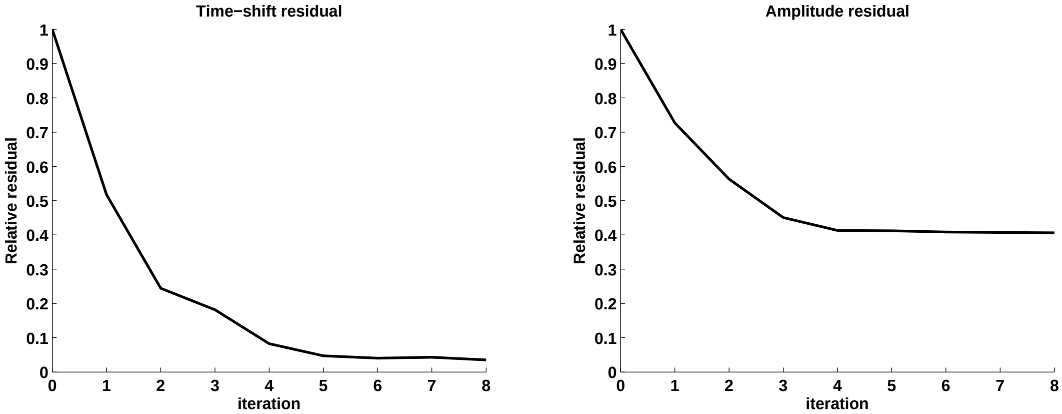

Figure 9. Reduction of residuals for all producers.

streaks. Plots of the residuals with respect to time-shift and amplitude in **Fig. 9** show that the iteration converges very fast (after 4–5 iterations). Results after eight iterations, updating all basis functions, are shown in **Fig. 7**. The updated permeability field is in general closer to the reference, and the realism of the permeability field is not degraded by the data integration process. **Figure 10** shows a comparison of the initial and final match of the fractional-flow curves for the wells with lowest initial, highest initial, and highest final mismatch (wells P3, P6, and P6, respectively). Overall, the match to the production data is quite satisfactory.

To test the robustness of our work reduction strategy we use a 5×5 test matrix with $x\%$ dynamical and $y\%$ initial update for $x, y = 0, 25, \ldots, 100$. **Figure 11** shows the reduction in residuals and permeability discrepancy after eight iterations. Judging from the amplitude residual and the permeability discrepancy, the data are well matched for all parameters $x, y >$

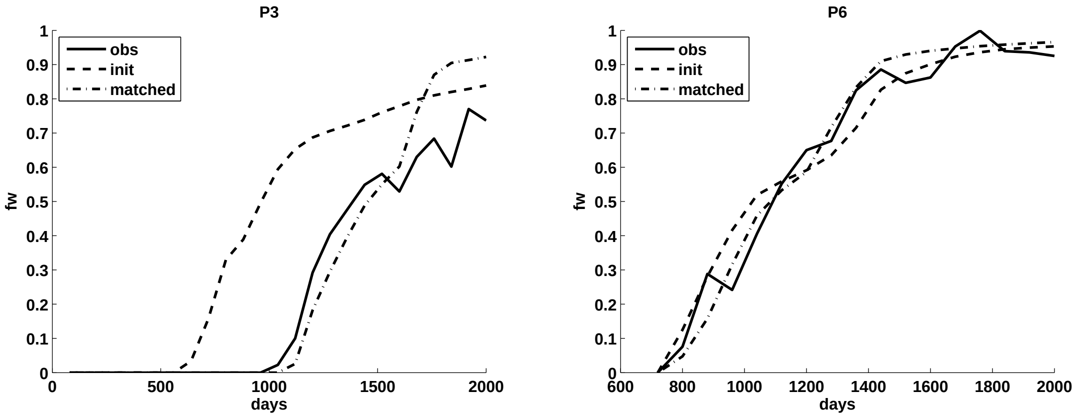

Figure 10. Fractional-flow curves for water for producers P3 (north-east), and P6 (east).

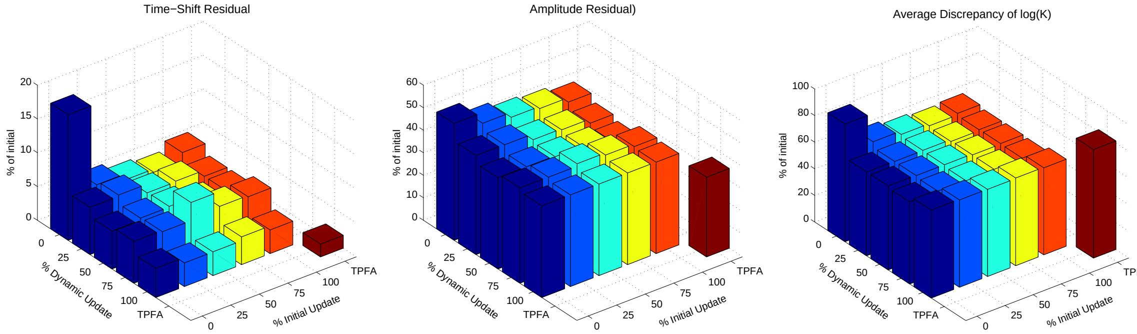

Figure 11. Robustness of selective work reduction.

0, and the quality of the match does not seem to decline dramatically compared with the TPFA solver. The time-shift residuals for MsMFEM are somewhat higher than for TPFA. The exception is $(x, y) = (0, 0)$, for which there is a significant decay in the quality of the match, in particular for the time-shift residual. **Figure 7** shows the resulting permeability field for $(x, y) = (0, 0)$ and $(100, 100)$. Even though the reduction in the residuals is significantly lower with no updating, the realism of the resulting permeability field seems as good as for full updating.

To explain the variations with respect to x and y, we consider the sensitivities. As seen in **Fig. 12**, the sensitivities are quite distinct and localized in channels (see also **Fig. 8**). The permeability field will typically change significantly from one iteration to the next in these channels, and failing to update the corresponding basis functions will lead to inaccurate results. However, for $x, y > 0$, our method seems to be able to select and update the basis functions contributing most to the production curves, see **Fig. 12**. The localized nature of the sensitivities makes it easier to cover the high-sensitivity areas with updated basis functions for quite low percentage values for x and y. In addition, the smoothing of permeability changes induced by the regularization in **Eq. 21** will also counteract the effect of sharp changes.

Finally, we emphasize that the reduction in runtime mainly will result from reducing the percentage of dynamically updated basis functions. It is therefore row $(x, 100)$ in the matrix

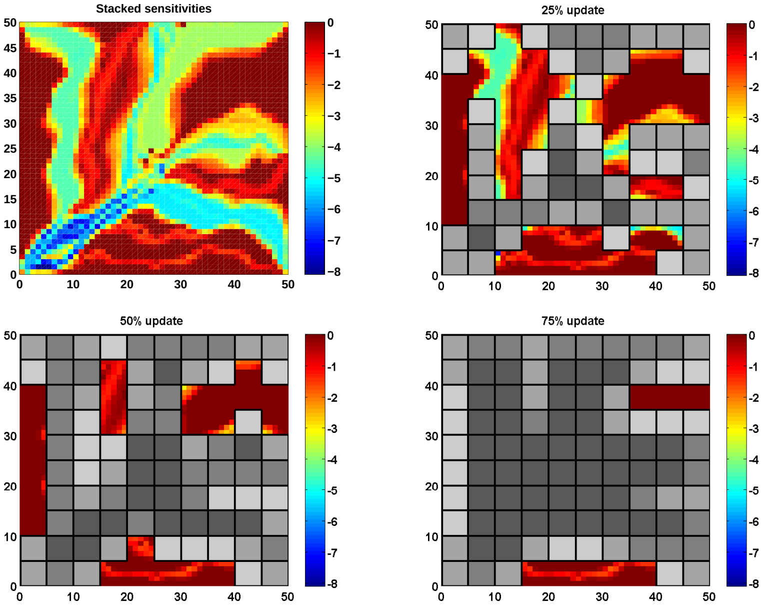

Figure 12. Stacked time-shift sensitivities (upper left). The three last figures show the coverage of updated basis functions for 25%, 50% and 75% updating. The four different levels of gray indicate the number of updated basis functions that cover each coarse grid block (brightest = 1, darkest = 4).

in **Fig. 11** that is of main interest when considering efficiency. In other words, we can avoid strategies in the matrix that give less stable results, and still get the intended speedup of the inversion process.

4.4. Multiple Equiprobable Realizations. We assume a multivariate Gaussian prior distribution for a 2-D reservoir model given on a 50×50 grid. As in the previous example, the prior distribution has long diagonal correlation length. The reference permeability field for this case is drawn from a slightly different multi-Gaussian distribution, see **Fig. 13**. The reservoir is produced from a five-spot pattern with an injector in the center and producers in the corners. As above, we assume quadratic relative permeability curves with zero residual oil and water saturations and end-point mobility ratio $M_{\text{end}} = 0.5$. For the MsMFEM pressure solver we construct a uniform 10×10 coarse grid such that each coarse block contains 5×5 fine cells.

Synthetic dynamic data were given by the fractional-flow curves obtained from the reference permeability field using the streamline method with a standard two-point pressure solver. To demonstrate the robustness of the generalized travel-time inversion, we match the observed data starting from a set of 25 permeability realizations of the assumed prior distribution. Here the permeability in each cell is treated as an adjustable parameter, giving a total of 2500 unknown parameters to be estimated for each realization.

Figure 13 shows three of the initial realizations and the corresponding matches after six iterations. The three matched permeability realizations are in closer agreement with the reference permeability field; a unique solution is not obtained since the data integration is ill-posed. **Figure 14** shows a comparison of the initial and final match of the fractional-flow

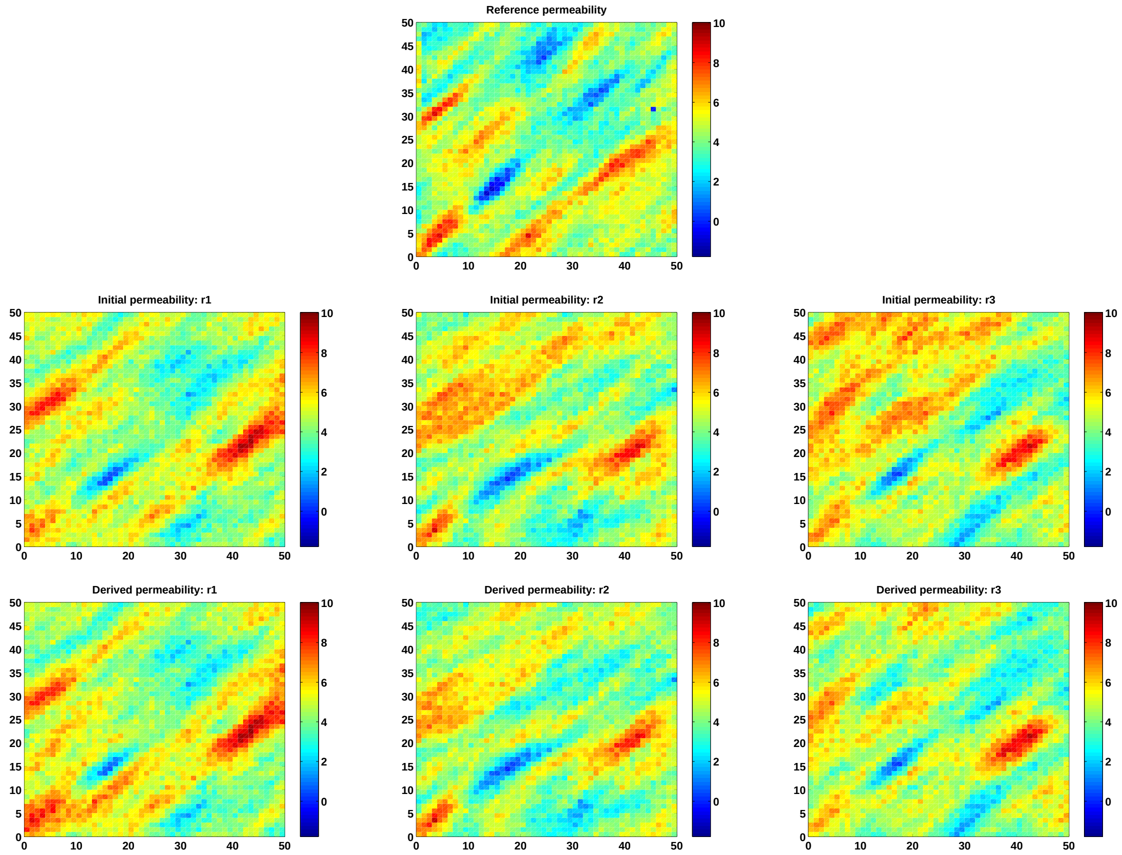

Figure 13. The plots show $\ln(k)$ for the reference permeability field (top) and initial (second row) and matched (third row) permeability fields for three different realizations.

Table 4. Mean and standard deviation of percentage reduction in misfit for time-shift residual (T) and amplitude residual (A).

	Misfit mean %		Std.dev. %	
Solver	T	A	T	A
Initial	100.0	100.0	100.0	100.0
TPFA	8.5	39.9	5.0	16.5
MsMFEM	7.8	38.9	4.6	17.6

curves for the set of permeability realizations for the four production wells. The time-shifts are obviously reduced considerably, and the amplitude is to some extent improved; also indicated in **Table 4**. Overall, the match to the production data is quite satisfactory.

Next we apply the same 5×5 test matrix for the set of realizations described above and measure the mean and standard deviations in the permeability discrepancy (**Eq. 25**) and the reduction in time-shift and amplitude residuals. **Figure 15** shows the result of the test matrix compared with results obtained with the TPFA solver. The data are well matched for all parameters x and y, and the quality of the history match does not seem to decline

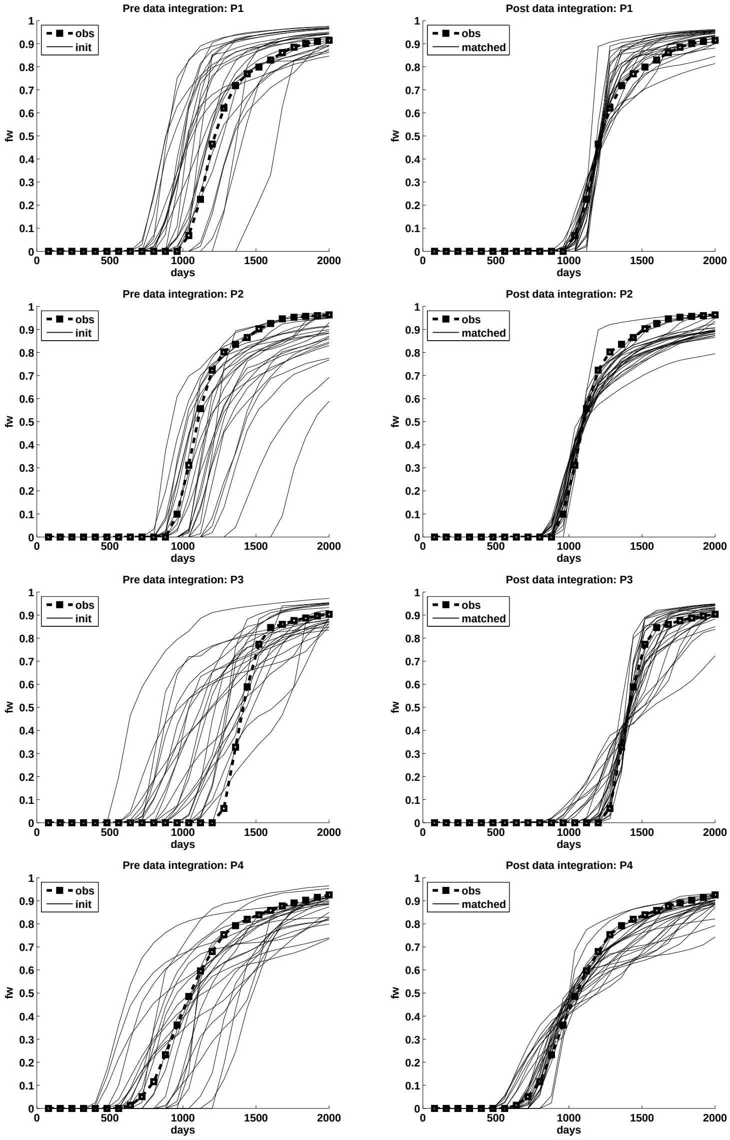

FIGURE 14. Fractional-flow curves prior and post data integration.

significantly compared with the TPFA solver even though a slight increase is observed by updating fewer basis functions.

4.5. History Matching a Full 3-D Geomodel. In (*Stenerud et al.*, to appear) we demonstrated the integration of dynamic fractional-flow data into a high-resolution 3-D geomodel with more than one million cells. We will now revisit this example and discuss the accuracy, robustness, and efficiency of our data-integration strategy in some more detail. In particular, we show that by combining generalized travel-time inversion with our modified streamline formulation and the selective work reduction, the computational challenging task of integrating data into a million-cell model can be surmounted in remarkable short time using a standard desktop or laptop computer.

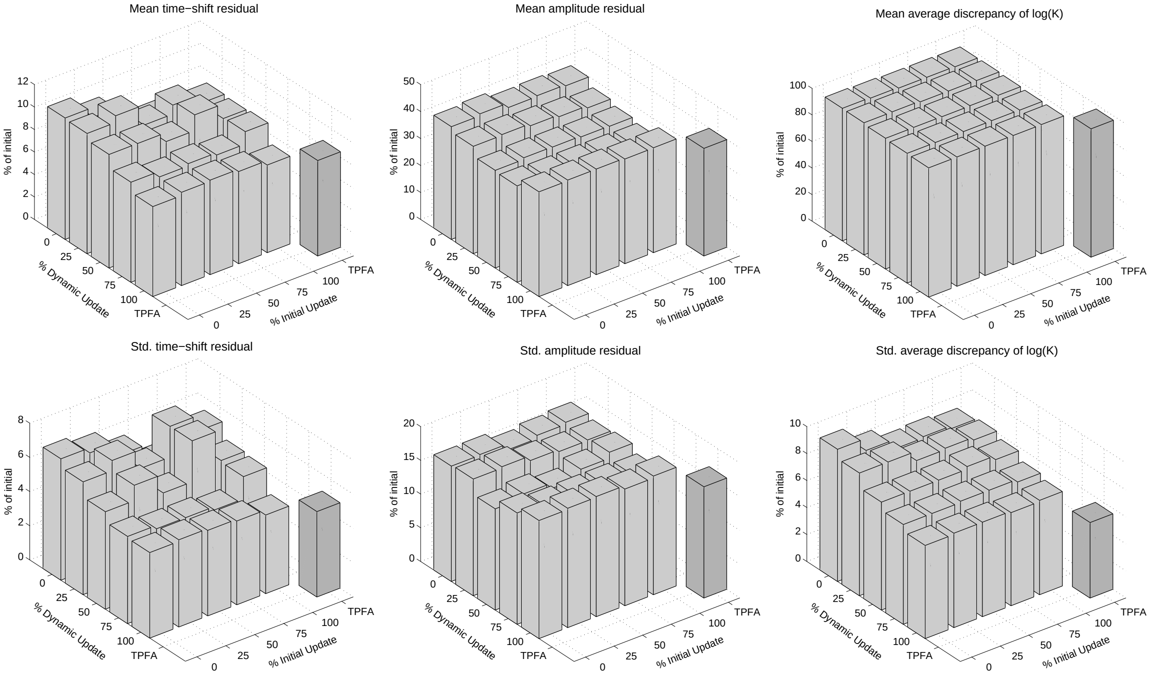

FIGURE 15. Mean (upper row) and standard deviation (lower row) of the reduction in time residuals, amplitude residuals and permeability discrepancies for a set of 25 permeability realizations.

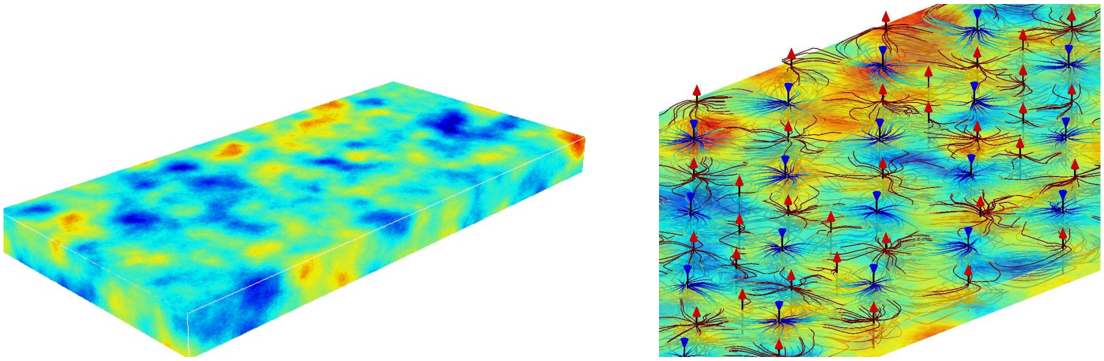

FIGURE 16. Permeability field and streamlines for the million-cell 3-D model.

The geomodel consists of a Cartesian fine grid with $256 \times 128 \times 32$ cells, which gives a total of 1 048 576 active cells, each of size $10 \times 10 \times 2$ m. We form a uniform $32 \times 16 \times 8$ coarse grid in which each block consists of $8 \times 8 \times 4$ fine cells. The permeability is log-normally distributed with a mean of 2.2 mD, a minimum of 0.017 mD and a maximum of 79.5 mD (see **Fig. 16**). The correlation length in the x- and y-directions is about 270 meters, and about 90 meters in the z-direction. The flow is described by the standard two-phase model with quadratic relative permeability curves and an end-point mobility ratio of $M_{\text{end}} = 5$.

The production history consists of 2475 days of fractional-flow data from the 69 producers, each operating with a constant rate fulfilling the total voidage rate induced by 32

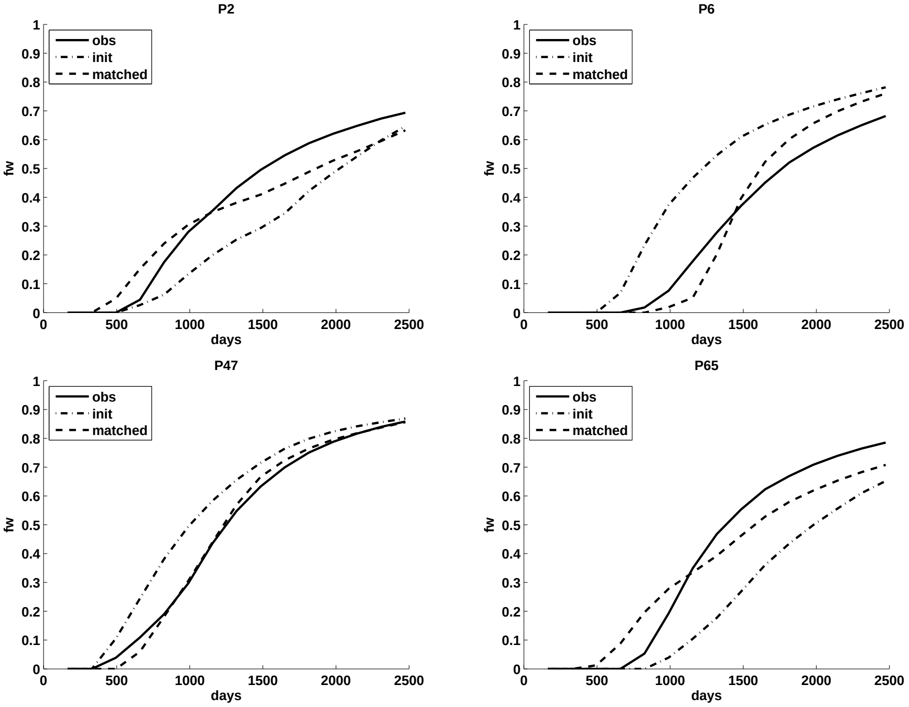

FIGURE 17. Match of fractional-flow curves for four of the 69 production wells included in the history match of the geologic model using the MsMFEM pressure solver.

water injectors operating with constant total reservoir-volume rate of 1609 bbl/day. For all simulations we used 15 pressure steps of 165 days. Good accuracy of the production curves required about 500 000 in the original streamlines method and about 50 000 streamlines for the modified formulation. Compared with the million-cell SPE 10 model in Section 3.2 this is a modest reduction. However, the current model has twenty times as many wells and is less dominated by heterogeneity structures compared with the SPE 10 model, in particular the bottom fifty fluvial layers.

An initial permeability model was generated using sequential Gaussian simulation (*Deutsch and Journel*, 1998) and conditioning on the permeability values in the well-blocks of the reference model. As reported in (*Stenerud et al.*, to appear), the misfit in time shift and amplitude had dropped appreciably after 5–6 iterations and the two pressure solvers gave almost identical derived permeability fields that both reserve the geologic continuity and the initial geologic features to the maximum possible extent. **Figure 17** shows initial and matched production curves for four producers.

First, we investigate to what extent the use of the modified streamline formulation and the multiscale pressure solver improves the computational efficiency of the data integration. As an example of a personal workstation, we use a recent commodity PC with a 2.4 GHz Intel Core 2 Duo processor with 4 Mb cache and 3 Gb memory. The TPFA pressure solver was compiled with full optimization and with multicore support for the underlying AMG linear solver. For MsMFEM, we were not able to optimize the underlying direct solver used to compute basis functions, nor did we exploit the parallelism of the Core 2 Duo processor.

Table 5. Reduction in percent in time-shift (T) and amplitude (A) residuals, and reduction in average discrepancy in log permeability ($\overline{\Delta \ln k}$). Runtimes are measured on a workstation PC with a 2.4 GHz Intel Core 2 Duo processor with 4 Mb cache and 3 Gb memory.

		Misfit			CPU-time (wall clock)		
Solver	O/M	T	A	$\overline{\Delta \ln k}$	Total	Pressure	Transport
Initial	—	100.0	100.0	0.821	—	—	—
TPFA	O	8.9	53.5	0.806	64 min	33 min	28 min
TPFA	M	9.6	50.4	0.806	39 min	30 min	5 min
MsMFEM	O	11.2	47.3	0.812	43 min	7 min	32 min
MsMFEM	M	10.4	45.4	0.828	17 min	7 min	6 min

Table 5 reports computational times (and reduction in misfit) using the TPFA and the MsMFEM pressure solvers in combination with the original and the modified streamline method. Here the total simulation time includes time for inversion, input/output, and seven forward simulations, each with fifteen pressure steps. Similarly, we report the total time for the pressure solves and the transport solves (including mappings and tracing of streamlines). Using the modified streamline method to reduce the number of streamlines from 500 000 to 50 000 reduced the time for the transport solves by 80% with negligible loss in accuracy. In (*Stenerud et al.*, to appear), we showed that reducing (or eliminating) the dynamical updates has almost no effect on the quality of the derived match for unfavorable mobility ratios. For MsMFEM, we used no initial and no dynamical updates and were thereby able to reduce time for pressure solves by about 80%, giving a significant reduction in the total runtime. Altogether, this meant that the full data integration could be performed in an impressive runtime of 17 minutes on a workstation PC!

To test the robustness of our selective work reduction, we apply the same 5×5 test matrix as in the two previous examples. For the strategies involving $x = 0\%$ dynamical update, no sensitivities are required to determine which basis functions to update. We therefore present results for both full dynamic update and no update during the first flow simulation for $x = 0\%$. **Figure 18** shows the reduction in residuals and permeability discrepancy after six iterations. Judging from the residuals and the permeability discrepancy, the data are well matched for all parameters, and the quality of the match is similar as for the TPFA solver. Altogether, the results indicate that the history-matching procedure is stable with respect to the selective work-reduction strategies.

Finally, we have tested how the speed-up for pressure solves for MsMFEM versus TPFA scales for the different work reduction strategies. To compare both solvers on more equal terms, we apply a somewhat older laptop PC with a 1.7 GHz Intel Pentium M processor so that both TPFA and the MsMFEM pressure solvers can be run with full optimization. Further, we are running all strategies with full dynamical update for the first flow simulation (because no sensitivities are available yet), except the strategies involving 0% dynamical update. Those strategies are run both with no and full dynamic updating during the first flow simulation. **Figure 19** shows the speed-up for the different combinations of updating strategies. The corresponding reduction in residuals and permeability discrepancy are shown in **Fig. 18**. As expected, the reduction in dynamic updates gives the greatest contribution

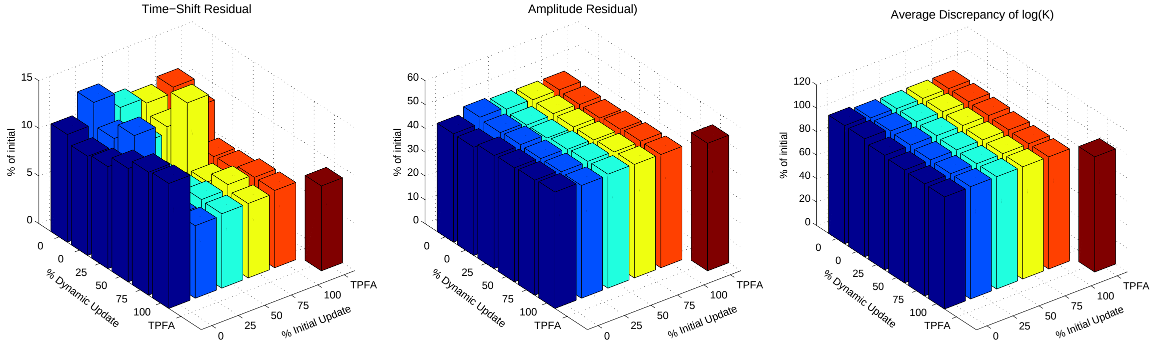

Figure 18. Robustness of selective work reduction. For 0% dynamical update, the first row corresponds no dynamic update during the first forward simulation and the second row to full update during the first simulation.

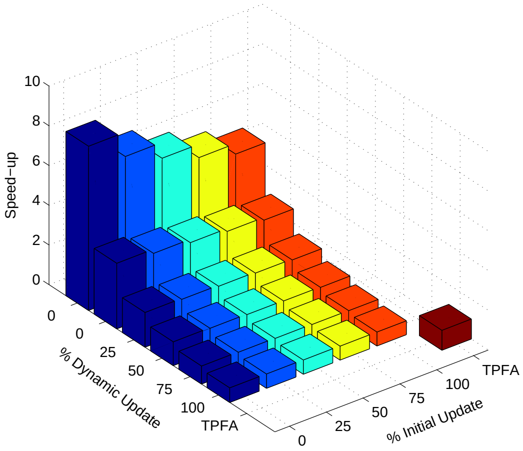

Figure 19. Speed-up matrix for time spent on pressure solves during the data integration. For 0% dynamical update, the first row corresponds no dynamic update during the first forward simulation and the second row to full update during the first simulation. Runtimes are measured on a 1.7 GHz Intel Dothan Pentium M processor with 2 Mb cache and 1.5 Gb memory

to the speed-up. With no updates, what so ever, the total data integration took 36 minutes on the laptop PC.

Finally, we notice that the memory requirements for the MsMFEM pressure solver are quite low and this solver could easily have been run on far larger models, as opposed to TPFA, for which the memory requirements of the AMG linear solver will rapidly limit the model sizes that can be run on a workstation or laptop PC.

5. Summary and Conclusions

A novel approach to data integration using multiscale-streamline simulation and analytic sensitivities is presented. There are four key components in our proposed approach:

(1) Inverse modeling by generalized travel-time inversion with quasilinear properties.
(2) Production-response sensitivities calculated analytically along streamlines.
(3) A modified streamline method that greatly reduces the mass-balance errors when simulating large and complex reservoir models using few streamlines.
(4) An efficient multiscale mixed finite-element method is applied as pressure solver, where the multiscale basis functions are recomputed selectively based on sensitivities.

The power and utility of our proposed approach was demonstrated using two 2-D models and a full field-scale geomodel consisting of more than a million grid cells. Starting with a prior subsurface model, production data were integrated using a generalized travel time inversion. The resulting permeability changes were found to be reasonable and geologically realistic (i.e., consistent with the initial geological model). For the one-million 3-D problem, the entire inversion process took about seventeen minutes using a commodity workstation PC. The very efficient forward simulation and sensitivity computations may generally enable history-matching of models with a large number of cells and/or a large number of (plausible) model realizations.

Altogether, we have presented very versatile method for integrating dynamic data into high-resolution subsurface models. The inversion method is applicable, with small modifications, to more general grid formats, as will be reported in (*Stenerud et al.*, submitted). We believe that using a MsMFEM pressure solver will prove particularly useful on complex grids, since the multiscale formulation gives a natural and automatic way of upscaling the grid in the pressure solver to speed up the forward simulation. Moreover, on highly skewed grids (e.g., corner-point grids), MsMFEM uses an accurate multipoint flux-approximation scheme to compute basis functions and therefore gives better spatial accuracy and more accurate predictions of flow than for a standard two-point method, see (*Aarnes et al.*, to appear).

Finally, we remark that the generalized travel-time inversion method has been extended to compressible three-phase flow, so that fractional-flow curves and gas-oil ratios are jointly incorporated (*Cheng et al.*, 2006). Moreover, data integration using streamline sensitivities has also been addressed for cases including gravity, changing field conditions, and fractured reservoirs (*He et al.*, 2002; *Al-Harbi et al.*, 2005).

Acknowledgments

The research of Stenerud was funded by the Uncertainty in Reservoir Evaluation (URE) program at the Norwegian University of Science and Technology. The research of Kippe and Lie was funded by the Research Council of Norway under grant number 152731/S30, 152732/S30 and 158908/I30.

References

Aarnes, J. E. (2004), On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation, *Multiscale Model. Simul.*, *2*(3), 421–439.

Aarnes, J. E., V. Kippe, and K.-A. Lie (2005), Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels, *Adv. Water Resour.*, *28*(3), 257–271.

Aarnes, J. E., S. Krogstad, and K.-A. Lie (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.

Aarnes, J. E., S. Krogstad, and K.-A. Lie (to appear), Multiscale mixed/mimetic methods on corner-point grids, *Comput. Geosci.*

Al-Harbi, M., H. Cheng, Z. He, and A. Datta-Gupta (2005), Streamline-based production data integration in naturally fractured reservoirs, *SPE J.*, *10*(4), 426–439.

Batycky, R. (1997), A three-dimensional two-phase field-scale streamline simulator, Ph.D. thesis, Stanford University, Dept. of Petroleum Engineering.

Bear, J. (1972), *Dynamics of Fluids in Porous Media*, American Elsevier, New York.

Chen, A., and T. Hou (2002), A mixed multiscale finite element method for elliptic problems with oscillating coefficients, *Math. Comp.*, *72*(242), 541–576.

Cheng, H., A. Datta-Gupta, and Z. He (2005), A comparison of travel-time and amplitude matching for field-scale production-data integration: Sensitivity, nonlinearity, and practical implications, *SPE J.*, *10*(1), 75–90.

Cheng, H., D. Oyerinde, and A. Datta-Gupta (2006), Compressible streamlines and three-phase history matching, in *SPE/DOE Symposium on Improved Oil Recovery*, Tulsa, Oklahoma, USA, SPE 99465.

Christie, M., and M. Blunt (2001), Tenth SPE comparative solution project: A comparison of upscaling techniques, *SPE Reservoir Eval. Eng.*, *4*(4), 308–317, url: www.spe.org/csp.

Deutsch, C., and A. Journel (1998), *GSLIB Geostatistical Software Library and User's Guide*, Oxford University.

He, Z., S. Yoon, and A. Datta-Gupta (2002), Streamline-based production data integration with gravity and changing field conditions, *SPE J.*, *7*(4), 423–436.

Hohl, D., E. Jimenez, and A. Datta-Gupta (2006), Field experiences with history matching an offshore turbiditic reservoir using inverse modeling, in *SPE Annual Technical Conference and Exhibition*, San Antonio, Texas, USA, SPE 101983.

Holden, H., and N. Risebro (2002), *Front Tracking for Hyperbolic Conservation Laws*, *Applied Mathematical Sciences*, vol. 152, Springer, New York.

Kippe, V., H. Hægland, and K.-A. Lie (2007), A method to improve the mass balance in streamline methods, in *SPE Reservoir Simulation Symposium*, Houston, Texas, USA, SPE 106250.

Kippe, V., J. E. Aarnes, and K.-A. Lie (to appear), A comparison of multiscale methods for elliptic problems in porous media flow, *Comput. Geosci.*

Matringe, S., and M. Gerritsen (2004), On accurate tracing of streamlines, in *SPE Annual Technical Conference and Exhibition*, Houston, Texas, USA, SPE 89920.

Paige, C., and M. Saunders (1982), LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Transactions on Mathematical Software*, *8*(1), 43.

Pallister, I., and D. Ponting (2000), Asset optimization using multiple realizations and streamline simulation, in *SPE Asia Pacific Conference on Integrated Modelling for Asset Management*, Yokohama, Japan, SPE 59460.

Pollock, D. (1988), Semi-analytical computation of path lines for finite-difference models, *Ground Water*, *26*(6), 743–750.

Ponting, D. (1998), Hybrid streamline methods, in *SPE Asia Pacific Conference on Integrated Modelling for Asset Management*, Kuala Lumpur, Malaysia, SPE 39756.

Qassab, H., M. K. R, Pavlas, N. Afaleg, H. Ali, A. Kharghoria, Z. He, S. Lee, and A. Datta-Gupta (2003), Streamline-based production data integration under realistic field conditions: Experience in a giant Middle-Eastern reservoir, in *SPE Annual Technical Conference and Exhibition*, Denver, Colorado, USA, SPE 84079.

Stenerud, V., V. Kippe, and K.-A. Lie (submitted), Generalized travel-time inversion on unstructured grids.

Stenerud, V., V. Kippe, K.-A. Lie, and A. Datta-Gupta (to appear), Adaptive multiscale streamline simulation and inversion for high-resolution geomodels, *SPE J.*

Stüben, K. (2000), *Algebraic Multigrid (AMG): An Introduction with Applications*, Academic Press, guest appendix in the book *Multigrid* by U. Trottenberg and C.W. Oosterlee and A. Schüller.

Vasco, D., S. Yoon, and A. Datta-Gupta (1999), Integrating dynamic data into high-resolution models using streamline-based analytic sensitivity coefficients, *SPE J.*, *4*(4), 389–399.

Vega, L., D. Rojas, and A. Datta-Gupta (2004), Scalability of the deterministic and bayesian approaches to production data integration, *SPE J.*, *9*(3), 330–338.

Yoon, S., A. Malallah, A. Datta-Gupta, D. Vasco, and R. Behrens (2001), A multiscale approach to production-data integration using streamline models, *SPE J.*, *6*(2), 182–192.