SPE 119132

Multiscale Mimetic Solvers for Efficient Streamline Simulation of Fractured Reservoirs

J.R. Natvig and B. Skaflestad, SINTEF ICT; F. Bratvedt and K. Bratvedt, Schlumberger; K.–A. Lie, SINTEF ICT;
V. Laptev and S.K. Khataniar, Schlumberger

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

Abstract

Advances in reservoir characterization and modeling have given the industry improved ability to build detailed geological models of petroleum reservoirs. These models are characterized by complex shapes and structures with discontinuous material properties that span many orders of magnitude. Models that represent fractures explicitly as volumetric objects pose a particular challenge to standard simulation technology with regard to accuracy and computational efficiency.

We present a new simulation approach based on streamlines in combination with a new multiscale mimetic pressure solver with improved capabilities for complex fractured reservoirs. The multiscale solver approximates the flux as a linear combination of numerically computed basis functions defined over a coarsened simulation grid consisting of collections of cells from the geological model. Here, we use a mimetic multipoint flux approximation to compute the multiscale basis functions. This method has limited sensitivity to grid distortions. The multiscale technology is very robust with respect to fine-scale models containing geological objects such as fractures and fracture corridors. The methodology is very flexible in the choice of the coarse grids introduced to reduce the computational cost of each pressure solve. This can have a large impact on iterative modeling workflows.

Introduction

Modern reservoir characterization methods and 3D geological modeling are leading the industry to routinely build very large and detailed geological models. These models currently may range in size from 10 to 100 million grid cells and are growing. This has resulted in a steadily increasing gap between flow simulation capability and the desire to build geologic-scale reservoir simulation models. In addition to sheer size, strong heterogeneity in the geological models may create computational problems. Geological models may use very small cells, have highly contrasting reservoir properties, and often have a lower proportion of active cells, which are widely distributed, thereby producing extremely complex hydraulic connectivity. Traditional finite-difference simulators were not designed to handle such models efficiently.

This is particularly true for fractured reservoirs, which are very difficult to manage and to optimize recovery for; see Bockel-Rebelle et al. (2005). About 60% of the world's conventional oil reserves and almost half of its gas reserves are contained in carbonate reservoirs, which tend to be more naturally fractured than sandstone reservoirs. To improve recovery factors, it is essential to have a thorough understanding of the depletion and displacement processes. Fractured reservoirs are complex geological structures in which fluids are stored in matrix blocks and flow occurs in the fractures. It is recognized that state-of-the-art simulation methods based upon dual-porosity descriptions may not be able to deliver sufficient resolution of the complex flow patterns that may develop when a fractured reservoir is produced. Several approaches (e.g., Matthäi et al. 2007) have therefore been taken to accurately describe fracture-fault systems on a grid-block scale, e.g., based upon complex gridding schemes in which fractures are represented explicitly either as volumetric grid cells or as lower-dimensional objects at the cell faces. The performance of current finite-difference simulators can drop significantly when detailed descriptions and complex matrix-fracture transport processes are introduced.

Traditional reservoir-modeling workflows have been deterministic, with a single "best effort" description of the reservoir and little or no quantitative evaluation of uncertainty in the data and its impact on predictions. More recently, geoscientists have taken advantage of increased computing power to generate many realizations to capture uncertainty in the static geological model. Discriminating between these multiple realizations requires dynamic simulation.

In the case of reservoirs where there is a substantial production history, each of the realizations must be simulated and the predicted reservoir response statistically compared with history; the realizations giving the closest match may then be selected for further study. In some cases, many realizations may give equally good matches to history. In this case, it may be necessary to acquire additional data to discriminate between them. Alternatively, several such "equally good" realizations may be used and the management decisions tested against each of them, giving a measure of uncertainty in the decision outcome. For example, if the decision is an infill drilling location, and every realization matching the history gives similar results for the infill location, then there is a high degree of confidence in that location.

The above issues result in a severe limitation on practical implementation of iterative modeling workflows because run times or cost of dynamic simulation on geological-scale models is still prohibitive using finite-difference simulators. Today, multiple realizations on the geological scale often have to be upscaled for evaluation using finite-difference simulators. Upscaling requires another, often overlooked, step of validation to ensure that the coarse model correctly represents the major geologic features that affect flow. There is no objective way yet to conduct or validate upscaling and hence it not only introduces an extra element of uncertainty but also complicates iterative workflows. Ideally, we would like to conduct high-resolution modeling and simulation at a uniform scale and on a desktop computer.

Streamline simulation has obvious speed advantages on large models and is steadily gaining acceptance as a key technology for processing large geological models as well as the more standard water flooding and history-matching applications (Lolomari et al. 2000; Grinestaff and Caffrey 2000; Kretz et al. 2004; Moreno et al. 2004; Cheng et al. 2004). To be able to simulate flow efficiently on increasingly detailed geological models, it is vital that the performance of the flux field computation required by the streamline algorithm be improved. With a multiscale mimetic pressure solver, modern streamline technology is able to meet this need from the industry, as suggested by Aarnes et al. (2006a). In this paper, we show how this novel simulation technology can be used on geological models with explicit fractures with limited or no upscaling.

Multiscale-Streamline Simulation

Multiscale methods are based upon a hierarchical grid structure consisting of a coarse simulation grid and an underlying fine-scale grid containing the reservoir heterogeneities. The global flow problem is solved on the coarse simulation grid using special basis functions that have sub-grid resolution. The basis functions are constructed by solving local flow problems on the fine-scale grid and can be used to reconstruct a conservative approximation of the fine-scale fluxes (and pressures). Several multiscale methods have been developed for the simulation of highly heterogeneous geological models, but they have so far been based upon either simplifying assumptions with respect to flow physics (two-phase, incompressible flow, etc.), e.g., Aarnes et al. (2008), or simplified geometry (Cartesian grids or the absence of fractures and faults), e.g., Hajibeygi et al. (2008). In a companion paper (Krogstad et al. 2009), we present a new multiscale method for three-phase black-oil flow on geological models with industry-standard complexity. For completeness, we will outline this method briefly below before we explain two potential usages within streamline simulation. To simplify the presentation, we will henceforth disregard gravity and assume noflow boundary conditions and that there are no wells in the interior of the domain. A more detailed description of how to incorporate wells and general boundary conditions is given by Skaflestad and Krogstad (2008).

Mathematical Model and Subgrid Discretization. We start by introducing the mathematical model and the mimetic discretization (Brezzi et al. 2005) to be used on the fine-scale grid. Streamline methods are based on a sequential solution of the equations written using a fractional flow formulation in which the pressure equation reads

$$c_t \frac{\partial p}{\partial t} + \nabla \cdot \vec{u} - \zeta \vec{u} \cdot \mathsf{K}^{-1} \vec{u} = q, \qquad \vec{u} = -\mathsf{K}\lambda \nabla p. \tag{1}$$

Here the total mobility $\lambda = \sum \lambda_\ell$ and all other saturation-dependent parameters are evaluated using saturations from the previous time step $n-1$; K is the absolute permeability; ζ is shorthand for $\sum c_\ell f_\ell$, where $f_\ell = \lambda_\ell / \lambda$ is the fractional flow; $q = \sum q_\ell / b_\ell$ is the total source; and c_ℓ and c_t are the phase and total compressibilities. Introducing a backward discretization in time, linearizing Eq. 1, and introducing an iteration parameter ν, the pressure equation Eq. 1 can be written in the semi-discrete form

$$c_t \frac{p^n_\nu - p^{n-1}}{\Delta t} + \nabla \cdot \vec{u}^n_\nu - \zeta^n_{\nu-1} \vec{u}^n_{\nu-1} \cdot \mathsf{K}^{-1} \vec{u}^n_\nu = q, \qquad \vec{u}^n_\nu = -\mathsf{K}\lambda \nabla p^n_\nu. \tag{2}$$

The computational domain Ω is assumed to be discretized by a set of polyhedral cells. For a given cell E, let $\boldsymbol{u}_E$ be the vector of outward fluxes over the n_E faces of E, p_E the pressure at the cell center, and $\boldsymbol{\pi}_E$ the pressures at the cell faces. Similarly, let $\boldsymbol{N}_E$ be the matrix containing the normal vector of the cell faces multiplied by the corresponding face area and $\boldsymbol{X}_E$ be the matrix containing the vectors from the face centroids to the cell centroid. Introducing a transmissibility matrix $\boldsymbol{T}_E$, the fluxes and the two pressures can be related as follows:

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

For the mimetic method, $\boldsymbol{T}_E$ is generally a full matrix that can be constructed by imposing exactness for linear pressures. Hence, $\boldsymbol{T}_E$ can be written in the form $\boldsymbol{T}_E = |E|^{-1} \boldsymbol{N}_E \mathsf{K}_E \boldsymbol{N}_E^{\mathsf{T}} + \tilde{\boldsymbol{T}}_E$ (see Brezzi et al. 2005), where the symmetric positive-definite matrix $\tilde{\boldsymbol{T}}_E$ can be chosen arbitrarily as long as it satisfies $\tilde{\boldsymbol{T}}_E \boldsymbol{X}_E = \mathbf{0}$. This gives us a certain freedom in the discretization, and the mimetic method can be constructed such that it, e.g., coincides with either the two-point method or the mixed finite-element method with lowest-order Raviart–Thomas basis functions on a Cartesian grid. We note additionally that first-order spatial convergence of both the pressure and flux variables was established in **?** under suitable assumptions on the tensor and grid cell shapes.

To finish the discretization of Eq. 2, we let $\boldsymbol{u}$ denote the outward fluxes over the faces ordered cell-wise (in which interior faces appear twice with opposite signs), $\boldsymbol{s}$ the cell-wise saturations, $\boldsymbol{p}$ the cell pressures, and $\boldsymbol{\pi}$ the face pressures. We then get a hybrid system

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^{\mathsf{T}} - \boldsymbol{V}_{\nu-1}^{\mathsf{T}} & \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{D}^{\mathsf{T}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_\nu \\ -\boldsymbol{p}_\nu \\ \boldsymbol{\pi}_\nu \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{P}\boldsymbol{p}^{n-1} + \boldsymbol{q} \\ \boldsymbol{0} \end{bmatrix}, \tag{4}$$

which we solve until $\|\boldsymbol{p}_{\nu-1} - \boldsymbol{p}_\nu\|$ and $\|\boldsymbol{u}_{\nu-1} - \boldsymbol{u}_\nu\|$ are sufficiently small. The matrix $\boldsymbol{P}$ is diagonal with entry $(c_t|E|/\Delta t)$ for cell E. The matrix $\boldsymbol{B}$ is block diagonal with one block $(\lambda \boldsymbol{T})_E^{-1}$ for each cell E, and similarly, $\boldsymbol{C}$ contains blocks of $n_E \times 1$ vectors with all entries equal one. If $\boldsymbol{D}_\zeta$ and $\boldsymbol{D}_{\boldsymbol{u}}$ denote diagonal matrices with ζ^{n-1} and $\boldsymbol{u}_{\nu-1}$ on the diagonal, respectively, then the block matrix $\boldsymbol{V}_{\nu-1} = \boldsymbol{V}(\boldsymbol{u}_{\nu-1})$ is given by $\boldsymbol{B}\boldsymbol{D}_{\boldsymbol{u}}\boldsymbol{C}\boldsymbol{D}_\zeta$. Finally, each column of $\boldsymbol{D}$ corresponds to a unique face and has one or two unit entries (for boundary and interior faces, respectively) corresponding to the index/indices of the face in the cell-wise ordering. We resolve the discrete system Eq. 4 by means of a standard Schur complement reduction to a related system for the interface pressures, $\boldsymbol{\pi}$. Specifically, the above system of linear equations is reduced to

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{0} & -\boldsymbol{L}_{\nu-1} & -\boldsymbol{F}_{2,\nu-1} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{S}_{\nu-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_\nu \\ -\boldsymbol{p}_\nu \\ \boldsymbol{\pi}_\nu \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{P}\boldsymbol{p}^{n-1} + \boldsymbol{q} \\ \boldsymbol{F}_1^{\mathsf{T}}\boldsymbol{L}^{-1}(\boldsymbol{P}\boldsymbol{p}^{n-1} + \boldsymbol{q}) \end{bmatrix}, \tag{5}$$

where the component matrices are given by the relations

$$\begin{aligned} \boldsymbol{F}_1 &= \boldsymbol{C}^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{D}, & \boldsymbol{F}_{2,\nu-1} &= (\boldsymbol{C}^{\mathsf{T}} - \boldsymbol{V}_{\nu-1}^{\mathsf{T}})\,\boldsymbol{B}^{-1}\boldsymbol{D}, \\ \boldsymbol{L}_{\nu-1} &= (\boldsymbol{C}^{\mathsf{T}} - \boldsymbol{V}_{\nu-1}^{\mathsf{T}})\,\boldsymbol{B}^{-1}\boldsymbol{D} - \boldsymbol{P}, \quad \text{and} & \boldsymbol{S}_{\nu-1} &= \boldsymbol{D}^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{D} - \boldsymbol{F}_1^{\mathsf{T}}\boldsymbol{L}_{\nu-1}^{-1}\boldsymbol{F}_{2,\nu-1}. \end{aligned}$$

We note that $\boldsymbol{L}_{\nu-1}$, though dependent upon the solution at the previous iteration, is a diagonal matrix whose nonzero elements constitute one nonzero scalar value in each grid cell. Consequently, this matrix is trivially inverted. Moreover, once the interface pressure system $\boldsymbol{S}_{\nu-1}\boldsymbol{\pi}_\nu = \boldsymbol{F}_1^{\mathsf{T}}\boldsymbol{L}_{\nu-1}^{-1}(\boldsymbol{P}\boldsymbol{p}^{n-1} + \boldsymbol{q})$ has been resolved—typically by means of an iterative method amenable to nonsymmetric systems of simultaneous linear equations (e.g., “BiCGStab”)—then the cell pressure values $\boldsymbol{p}_\nu$ are obtained from the diagonal system

$$\boldsymbol{L}_{\nu-1}\boldsymbol{p}_\nu = \boldsymbol{P}\boldsymbol{p}^{n-1} + \boldsymbol{q} + \boldsymbol{F}_{2,\nu-1}\boldsymbol{\pi}_\nu.$$

Within a single grid cell, this process corresponds to expressing the cell pressure as a linear combination of source terms, the pressure value at the previous pressure step, and the interface pressures at the cell's boundary.

Finally, we derive the cell interface fluxes. Algebraically, this means resolving a block-diagonal system of linear equations given by

$$\boldsymbol{B}\boldsymbol{u}_\nu = \boldsymbol{C}\boldsymbol{p}_\nu - \boldsymbol{D}\boldsymbol{\pi}_\nu.$$

As the system's right-hand side constitutes a first-order approximation to the pressure gradient, this is the discrete counterpart to Darcy's law. The process of computing fluxes at a cell's interfaces is demonstrated in **Fig. 1**. Consider cell i whose pressure value is p_i. Denoting by $(\boldsymbol{B}^{-1})_{mn}$ the mn-th element of the inverse inner-product matrix $\boldsymbol{B}^{-1}$ restricted to cell i, the cell's outward flux across the interface connecting cell i to its neighbor j—the interface at which the pressure is π_5—may then be explicitly stated as

$$u_5 = \sum_{k=1}^{5} (\boldsymbol{B}^{-1})_{5k} \cdot (p_i - \pi_k). \tag{6}$$

Through this formula, the flux u_5 in Eq. 6 generally depends on *all* the pressure drops $p_i - \pi_k$. In the special case of a diagonal inner-product matrix, however, the flux depends only on the pressure drop from the cell center to the corresponding cell interface. We note furthermore that by construction, the outward flux from cell j across the connecting cell interface equals negative u_5. In practice, various arithmetic errors (i.e., round-off errors) may lead to tiny discrepancies and discontinuities in the flux field. The mimetic method is therefore usually accompanied by a postprocessing procedure that projects the mimetic fluxes onto a continuous flux field.

The Multiscale Mimetic Pressure Solver. Our multiscale pressure solver is a variant of the multiscale mixed finite-element method introduced by Chen and Hou (2003), in which we have replaced the inner-products by a mimetic approximation as discussed above and represent the multiscale basis functions in terms of a vector of fluxes on the fine grid inside each coarse block. This idea was first introduced by Aarnes et al. (2008) and is here extended to compressible, three-phase, black-oil models.

As explained above, the multiscale solver is based on a hierarchical two-level grid in which the blocks Ω_i in the coarse simulation grid consist of a connected set of cells from the underlying fine grid, on which the full heterogeneity is represented.

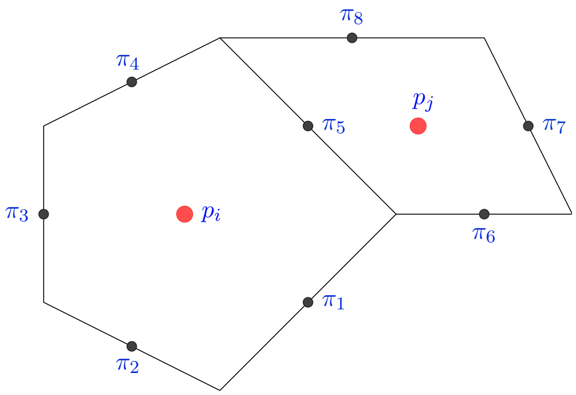


Fig. 1—Block and interface pressure values from which interface fluxes are derived.

In its simplest form, the approximation space consists of a constant approximation of the pressure inside each coarse block and a set of velocity basis functions associated with each interface between two coarse blocks. 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 solving

$$\vec{\psi}_{ij} = -\lambda \mathbf{K} \nabla p_{ij}, \qquad \nabla \cdot \vec{\psi}_{ij} = \begin{cases} w_i(x), & \text{if } x \in \Omega_i, \\ -w_j(x), & \text{if } x \in \Omega_j, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

in Ω_{ij} with $\vec{\psi}_{ij} \cdot \vec{n} = 0$ on $\partial\Omega_{ij}$. 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 distribute $\nabla \cdot \vec{u}$ over the block and produce a flow with unit average velocity 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.

Let $\boldsymbol{\psi}_{ij}$ denote the basis function constructed by solving Eq. 7 using the mimetic discretisation introduced above. To construct the coarse-scale system, we write the basis functions as $\boldsymbol{\psi}_{ij} = \boldsymbol{\psi}_{ij}^H - \boldsymbol{\psi}_{ji}^H$, where $\boldsymbol{\psi}_{ij}^H(E)$ is equal to $\boldsymbol{\psi}_{ij}(E)$ if $E \in \Omega_{ij} \setminus \Omega_j$ and zero otherwise, and $\boldsymbol{\psi}_{ji}^H(E)$ is equal to $-\boldsymbol{\psi}_{ij}(E)$ if $E \in \Omega_j$ and zero otherwise. These hybrid basis functions $\boldsymbol{\psi}_{ij}^H$ are now collected as columns in a matrix $\boldsymbol{\Psi}$. Then we introduce two prolongation operators $\boldsymbol{\mathcal{I}}$ and $\boldsymbol{\mathcal{J}}$ from blocks/coarse interfaces to cells/fine faces such that element ij equals one if block/coarse interface number j contains cell/face number i and zero otherwise. Then the global coarse-scale system reads,

$$\begin{bmatrix} \boldsymbol{\Psi}^{\mathsf{T}} \boldsymbol{B}_f \boldsymbol{\Psi} & \boldsymbol{\Psi}^{\mathsf{T}} \boldsymbol{C}_f \boldsymbol{\mathcal{I}} & \boldsymbol{\Psi}^{\mathsf{T}} \boldsymbol{D}_f \boldsymbol{\mathcal{J}} \\ \boldsymbol{\mathcal{I}}^{\mathsf{T}} (\boldsymbol{C}_f - \boldsymbol{V}_f)^{\mathsf{T}} \boldsymbol{\Psi} & \boldsymbol{\mathcal{I}}^{\mathsf{T}} \boldsymbol{P}_f \boldsymbol{\mathcal{I}} & \mathbf{0} \\ \boldsymbol{\mathcal{J}}^{\mathsf{T}} \boldsymbol{D}_f^{\mathsf{T}} \boldsymbol{\Psi} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ -\boldsymbol{p} \\ \boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\mathcal{I}}^{\mathsf{T}} \boldsymbol{P}_f \boldsymbol{p}_f^n \\ \mathbf{0} \end{bmatrix}, \tag{8}$$

where the subscript f indicates matrices from the fine-scale discretization. We remark that as Eq. 8 is of the same form as the fine-scale system of Eq. 4, this system may be solved using the Schur complement reduction strategy outlined in Eq. 5.Once Eq. 8 is solved and coarse-scale fluxes $\boldsymbol{u}$ have been computed by the coarse-scale equivalent to Eq. 6, the fine-scale fluxes can be obtained immediately as $\boldsymbol{u}_f = \boldsymbol{\Psi}\boldsymbol{u}$.

Whereas this formulation is efficient for incompressible and weakly compressible flows, a more complicated construction may be needed for strongly compressible cases if the coarse grid is not sufficiently fine to resolve compressible effects.

Applications to Streamline Simulation. The multiscale mimetic pressure solver has two very important attributes that can be used to improve the performance of streamline simulation when dealing with geological models for which the fracture systems or fracture corridors must be explicitly modeled. The first (multiscale) is the ability to efficiently compute the flux field and the second (mimetic approach) allows handling of distorted or non-orthogonal grids. Distorted grids are more the norm than the exception when the geological model is being constructed. Correct(ed) flow simulation on distorted grids again reduces the need to rebuild a grid for simulation purposes, further enhancing the speed of iterative modeling workflows. The mimetic method is a multipoint flux-approximation technique that is robust but can lead to larger number of unknowns. Hence, a multiscale method is preferred as a companion to maintain performance.

Considering the multiscale approach presented in this paper, there are two ways this can be applied within a streamline simulator:

- as a fast, approximate pressure solver that reduces the number of independent unknowns in the pressure equation;
- as a method for taking into account the impact of subcell heterogeneities that are not represented in the input grid.

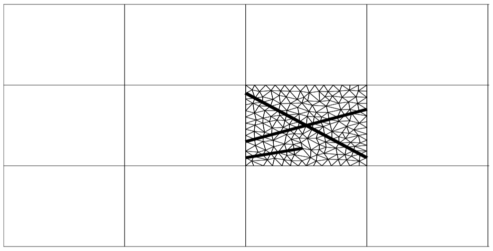


Fig. 2—A conceptual illustration of a coarse grid with a triangular refinement inside a single cell to accurately capture three fractures (thick lines).

The first approach is the classical multiscale modeling approach in which the primary input to the reservoir simulator is the fine-scale geologic grid and properties. All geologic features including fractures are modeled directly on the fine grid. The coarse grid, on which we seek to solve the pressure, can be constructed either automatically (e.g., by partitioning a logically Cartesian grid into rectangular blocks in index space), or with varying degrees of user control. The preliminary work done at the solver setup stage (computing the local solutions), is also used to decrease the number of independent unknowns in the pressure equation, so that they become associated with the coarse grid. The velocity field realized on the fine grid is a good approximation to the direct fine-grid solution. The objective is to solve the pressure equation more efficiently on a smaller linear system while allowing for some deterioration in accuracy.

We note that computing the local solutions has an additional cost and recomputing the local solutions may be needed for multiphase flow. This, however, is where we can use advances in computing to complement our methodology and maintain high performance. Every local problem is independent and is therefore a prime candidate for exploiting the capability of parallel computing inherent in multicore computers (and possibly also the massive parallel capability available in modern graphics cards). However, research may still be needed to derive a good load balancing, in particular for problems with many wells. Likewise, streamline generation and solving saturation equations along streamlines are naturally parallel tasks and multicore computing has already been harnessed for these tasks in a commercial streamline simulator. This can be particularly useful in fractured reservoir simulation, as considerable time can also be taken by the 1D finite-difference saturation solver owing to the high speed of flow along the streamlines.

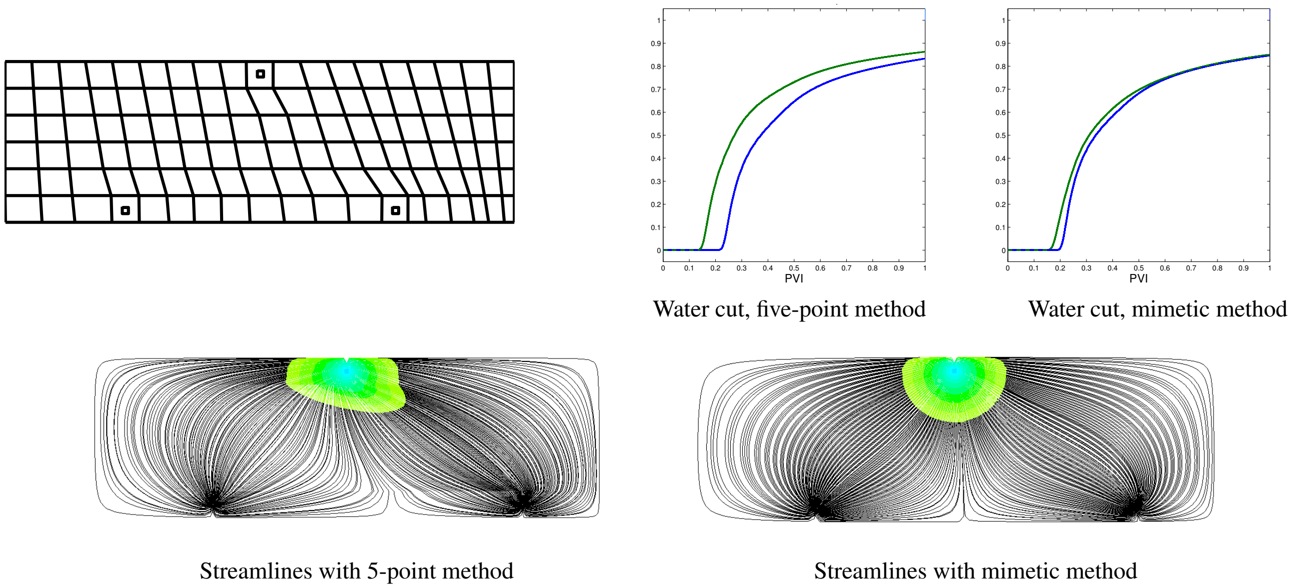
The second approach to applying the multiscale technique is to use two separate grid systems. The primary grid consists of the background geology on which a secondary grid or grid system representing the high-impact features like fractures is superimposed. The secondary grid is expected to have a resolution commensurate to the characteristics of the geologic features.

The primary grid can be used as a preliminary candidate for the coarse grid for the multiscale approach. The union of the primary and secondary grid must be constructed afterwards and automatically (if possible). The primary grid serves only as a preliminary candidate and may require modification to better model the discrete geologic objects. For example, it might be irrelevant to have higher grid resolution in areas away from fault corridors. Therefore, one can find the cells containing fractures and only refine them and possibly their neighbors. This way, one may combine the primary grid with locally refined structured or unstructured grids; see the conceptual illustration in **Fig. 2**. The mimetic discretization described above is well suited for both structured and unstructured grids containing polyhedral cells. The local solutions are used in assembling the matrices needed in solving the pressure equation, for constructing the detailed velocity field and tracing streamline inside the (coarse) cell. It is not necessary to store the whole fine grid explicitly; one only needs to obtain the relation between the cells in the primary grid, their refinement, and the composition of the coarse grid (which could be different from the primary grid). The cells having no refinement need no special care, no local solution, and no additional memory storage. They can be treated in the same way as they are treated in the mimetic method to minimize the storage and the work needed to assemble the matrices.

Numerical Results

In this section we will present three examples to highlight the utility of our new multiscale-streamline simulator. The first example is a classical and simple 2D test case designed to illustrate grid-orientation effects caused by skewed cells. In the second example, we show how the multiscale method can be used to resolve a detailed 2D model of hydraulic fractures on a much coarser grid. The third example is a 3D case with fracture corridors constructed from a subset of Model 2 of the Tenth SPE Comparative Solution Project (SPE 10; Christie and Blunt 2001).

Reduced Grid Effects. In the first example, we consider a rectangular reservoir with homogeneous and isotropic permeability of 500 mD (the porosity is 0.3). The reservoir is produced from a symmetric well pattern consisting of an injector with a bottomhole constraint of 600 bar and two producers with a rate constraint of 40 m^3/day (**Fig. 3**). We assume incompressible flow and use quadratic relative permeability curves. A total of $20,000$ days of production is simulated using 100 time steps, each of 200 days. As can be seen from the streamlines and water-cut curves in Fig. 3, use of the mimetic method strongly reduces the



Water cut, five-point method

Water cut, mimetic method

Streamlines with 5-point method

Streamlines with mimetic method

Fig. 3—Simple test case designed to compare grid-orientation effects in the mimetic and the standard five-point scheme. The flow pattern should be symmetric because of the symmetric well pattern.

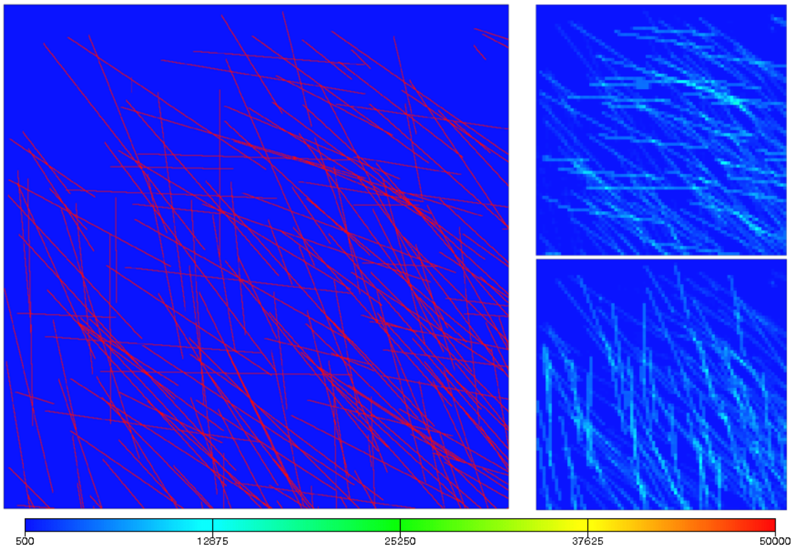


Fig. 4—The left plot shows the 2D system of high-permeability fracture corridors on a fine 800×800 grid.The right plots show the upscaled permeability field on a 80×80 grid with K_{xx} (top) and K_{yy} (bottom).

grid-orientation effects inherent in the standard five-point scheme.

A Dense System of Random Fracture Corridors. As our first example of a fractured media, we consider the 2D system of fractures presented in **Fig. 4**. The system of 200 linear fractures was constructed randomly and then discretized on a 800 × 800 Cartesian grid. The permeability value in the fractures is set to 50 darcies, whereas the permeability of the isotropic, homogeneous background field is 500 mD. The reservoir is produced using a quarter-five spot pattern with an injector in the top-left corner and a producer in the bottom-right corner. The grid was coarsened to 80 × 80 blocks using a standard flow-based diagonal upscaling procedure (prescribed unit pressure drop in the axial direction, no-flow conditions at the other boundaries); see the right-hand plots in Fig. 4.

To test the accuracy of our multiscale-streamline method, we compare saturations after 100 and 300 days. **Fig. 5** shows the reference water saturations computed directly on the 800 × 800 grid along with saturations obtained from the upscaled 80 × 80 grid with permeabilities computed by (i) standard flow-based diagonal upscaling, and (ii) geometrical upscaling. In the upscaled results, the fine-scale features are smeared away, and the injected fluid propagation is significantly slower than in the reference

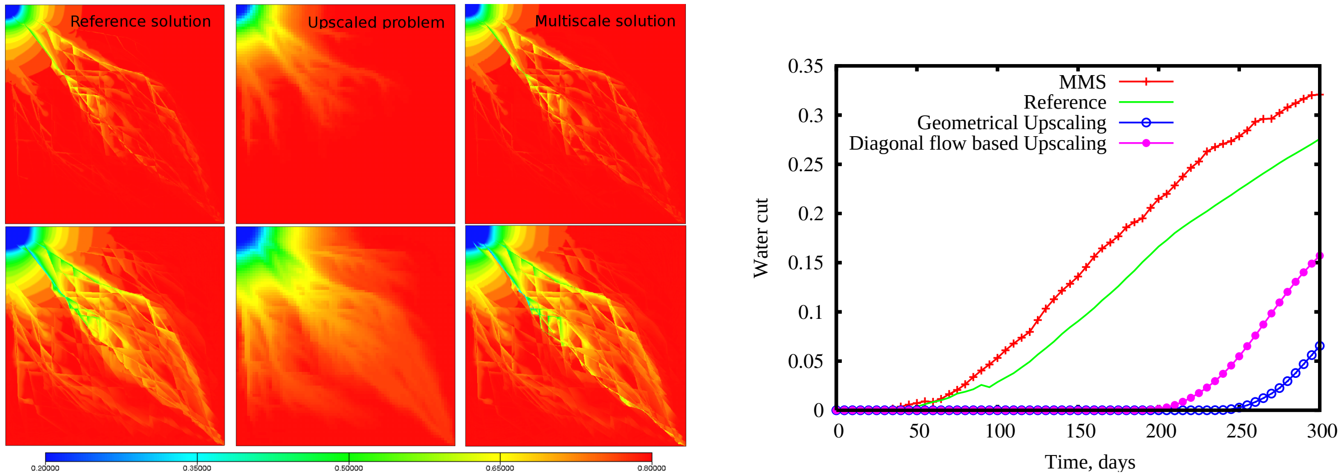


Fig. 5—The color plots show the saturation at t=100 days (top) and t=300 days (bottom). The left column is the reference solution on the 800×800 grid, the middle column is the upscaled problem (flow-based diagonal upscaling), and the right column is the multiscale-streamline solution. The graph shows the water cut: reference solution in green, multiscale solution in red, geometrical upscaling in blue, and flow-based diagonal upscaling in magenta.

case. The fractured system is rather dense in this test, and therefore the uniform 800×800 grid was chosen as the fine grid for the multiscale approach. For the coarse grid, we follow the suggestion made by Aarnes et al. (2006b). That is, we start with the 80×80 grid used in the upscaling and add extra coarse blocks to represent the fractures so that the union of all fine cells in fractures is split into coarse blocks by the boundaries between the blocks of the 80×80 upscaling grid. The results of this multiscale approach are shown in the right column of Fig. 5, from which we see that the multiscale-streamline solver is able to predict the fine-scale features of the reference solution quite well.

The water-cut curves are compared in the graph in Fig. 5. Although the water cut of the multiscale solution does not coincide with the reference solution, the correct breakthrough time and the qualitative behavior makes it more attractive than the water cuts from the two upscaling methods.

SPE 10 with Fracture Corridors. Next we consider a 3D example constructed from the upper seventeen (Tarbert) layers from Model 2 of the SPE 10 benchmark (Christie and Blunt 2001). This model is repeated periodically three times in the x-direction, giving a model with $180 \times 220 \times 17$ cells. Additionally, there are six fracture corridors added to the reservoir model (Montaron et al. 2007). We place one injector in the middle and producers at the corners of each periodicity part. Altogether there are three injectors and eight producers and, as in the original setup, we consider an injection period of 2,000 days.

The fracture corridors have again a permeability equal to 50 D. As in the previous example, we compute a reference solution by solving the flow directly on the fine-scale model. The initial grid for the multiscale test is 10 times coarser in the two horizontal directions. In addition, each connected fracture corridor also composes a single coarse block. The saturation fields in different layers 2,000 days after the start of injection and production are presented in **Fig. 6**. We see that the plots in the right column are quite similar to those in the middle, although not all flow paths are identical. To check how critical these differences are, we can compare some important cumulative characteristics, like oil production rates and water cut. Such comparisons are presented in **Figs. 7 and 8** and show good correspondence between the results; in particular, we see that the multiscale method is able to capture the correct water breakthrough in all wells. However, wells P7 and P8 have a noticable discrepancy in the water-cut curves. This discrepancy can be significantly reduced by adding extra coarse blocks that adapt to (and explicitly represent) the high-permeable fracture corridors, as was done in the 2D example above. Adding such media-adapted, and potentially highly irregular, grid blocks is straightforward and does not induce any complications in the multiscale solver; the coarse-grid partitioning can still be represented as a vector of cell indices and basis functions are still defined over connected collections of fine-grid cells. The interested reader is referred to Aarnes et al. (2006b) for a more detailed discussion of media-adapted coarse grids.

Computational Efficiency The computational efficiency of the multiscale-streamline formulation has been documented previously; see Aarnes et al. (2006a); Stenerud (2007); Stenerud et al. (2008). In particular, Stenerud (2007) reports how a highly optimized inhouse multiscale-streamline solver was able to compute the whole production history of the SPE 10 benchmark in 2 minutes and 22 seconds using a single core of a PC with a 2.4 GHz Intel® Core 2 Duo processor with 4 MB cache and 3 GB memory. In this simulation, the authors assumed incompressible flow, neglected gravity, and used a $5 \times 11 \times 17$ coarse grid with a standard two-point discretization of the local flow problems defining the basis functions.

Computational efficiency of the multiscale pressure solver requires the following three discrete capabilities: (i) an efficient linear solver for computing the basis functions, (ii) reuse of basis functions from one time step to the next, and (iii) efficient

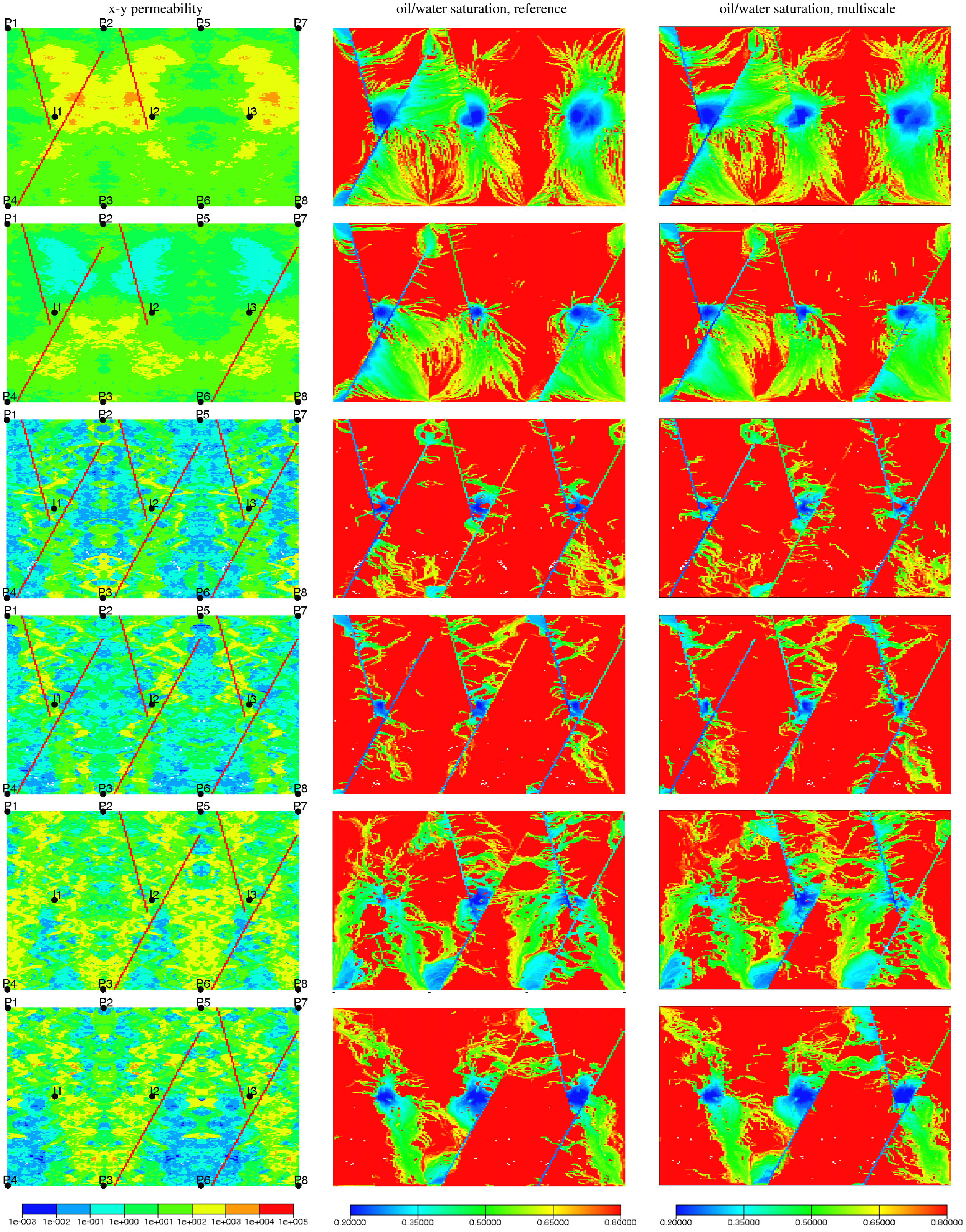


Fig. 6—Comparison of saturation fields in Layers 4, 6, 8, 10, 12, and 14 after 2,000 days for the 3D model with fracture corridors.

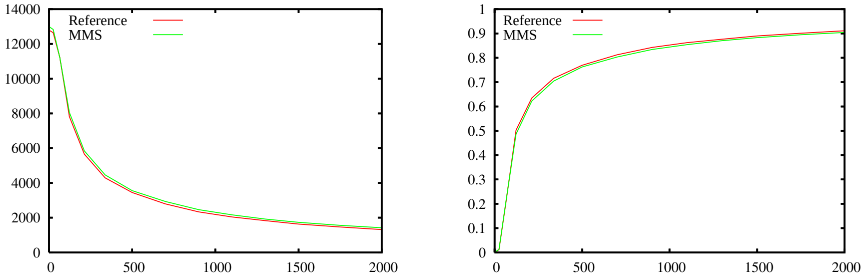


Fig. 7—Comparison of field oil-production rate in STB/day (left) and field water cut (right) for the 3D model with fracture corridors for 2,000 days of production: reference solution in red and multiscale solution in green.

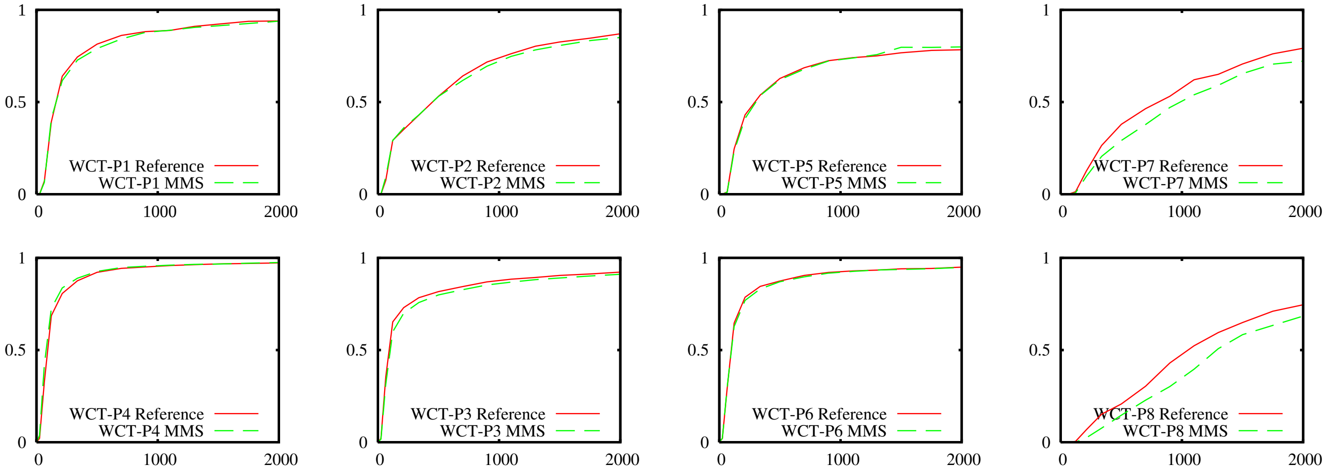


Fig. 8—The top row shows water-cut curves for producers along the upper edge (P1,P2,P5,P7) and the lower row the producers along the lower edge (P4,P3,P6,P8) for the 3D model with fracture corridors: reference solution in red and multiscale solution in green.

assembly and solution of the global coarse-scale flow problems. The prototype simulator used to run the numerical examples reported above has the latter two capabilities, but does not yet have a linear solver that is efficient for computing basis functions, i.e., solving a large number of small or medium-sized flow problems simultaneously. However, developing an efficient linear solver for the basis functions was considered outside the scope of this paper. Several approaches have been developed in flow-based upscaling; known approaches include reuse of symbolic factorization in direct solvers, gathering the local flow problems into larger systems, and using suitable iterative solvers.

Despite the shortcomings of our prototype implementation, we report run time comparisons for a set of full SPE 10 data simulations. To this end, we consider: (i) an upscaled version of the SPE 10 model, consisting of 30 × 110 × 17 cells; (ii) the original SPE 10 model; and (iii) a 3 × 3 repetition of the SPE 10 geological data (reusing the original SPE 10 geology nine times), rescaled to fill the volume of the original model. The three models have 56 thousand, 1.1 million, and 10 million cells, respectively. In **Table 1**, we report the time (in seconds) for initializing the problem, computing basis functions, assembling the global system, solving the pressure system and deriving the (fine-scale) flux field, computing saturation transport, and total time. To reach 2000 days, we used 13 nonuniform time steps. Model (i) was additionally run using 50 equally sized time steps. We furthermore note that the individual transport steps were parallelized using 16 cores. With an upscaling factor of $5 \times 5 \times 5$, the total simulation time for the multiscale-streamline solver is significantly lower than the corresponding fine-scale mimetic solver. Notice, however, that this solver is also a prototype, for which neither computing the inner product during the assembly process nor the linear solver have been optimized. For comparison, the run time using the highly optimized commercial simulator with a two-point pressure solver was 170 seconds for the original 1.1 M model. (Here, the number of degrees of freedom is approximately 1/3 compared with the mimetic pressure solver.)

In summary, our preliminary experiments indicate that the multiscale method has a clear potential for accelerating the pressure solution and that the advantage of using the multiscale solver increases with the number of time steps (because of reuse of basis functions) and with increasing model sizes. In particular, the memory consumption of the multiscale solver seems to scale much better than that of the AMG fine-scale solver.

TABLE 1—RUN-TIME COMPARISON, USING THE MIMETIC FINITE DIFFERENCE DISCRETIZATION FOR THE PRESSURE EQUATION, FOR THREE MODELS DERIVED FROM MODEL 2 OF THE SPE 10 BENCHMARK.

Model	Solver	Grid	Steps	Initialization	Basis	Assembly	Pressure+Flux	Transport	Total
56 k	AMG	30×110×17	13	3	—	26	96	2	129
			50	8	—	89	261	14	373
	M-S	6× 22×17	13	3	13	2	2	5	27
			50	8	11	2	4	18	44
1.1 M	AMG	60×220×85	13	46	—	525	1,787	38	2,424
	M-S	12× 44×17	13	46	350	27	14	45	514
10 M	AMG	180×660×85	13	470	—	4,803	25,538	398	31,401
	M-S	36×132×17	13	470	2,597	193	169	305	3,925

Conclusions

The multiscale approach has the ability to reconstruct a fine-scale velocity while taking into account the high flowrate in narrow fracture corridors, so that the streamlines constructed from the fine-grid velocity are denser in the fracture corridors. Consequently, the multiscale-streamline solution reproduces the small-scale features of the fine-grid solution. Although not discussed herein, our experience indicates that the quality (accuracy) of the multiscale results depends strongly on the geometry of the coarse blocks. Better results are obtained when some coarse blocks are devoted to the fracture corridors by following their geometry; see Aarnes et al. (2006b) for a more thorough discussion for the opposite case of impermeable objects.

Regardless of the promising results reported herein, flow simulation in fractured reservoirs remains a challenging task. One aspect making it difficult is that the fractures exist at a much smaller scale than the typical cell size in reservoir discretizations. They create effects that require a very fine grid to be taken into account by conventional flow simulators. If our purpose is explicit fracture treatment, we should be able to construct a local grid around the fractures and solve the equations governing the flow on such a grid. After achieving the smallest possible size of cells in the refinement, the population of properties in the cells containing fractures could be done using upscaling techniques. The refinement is limited by the manageable total number of cells. These limits can be improved by the multiscale approach. Also, the construction of the finest grid with the multiscale approach could be made much more flexible. Currently only the Cartesian refinement (not very suitable for fractures, leading to stair-similar approximations) is supported by our simulator. The local base functions in the multiscale approach could be constructed using this standard grid, but there is also a potential possibility of defining these functions on special grids well adopted and adjusted to the geometry of the fractures and thus minimizing the total number of cells and minimizing the errors in discretizing the geometry. This is a topic of future research.

Acknowledgments

The authors gratefully acknowledge financial support from the Research Council of Norway under grant number 174551/S30 and Schlumberger for permission to publish this paper.

Nomenclature

Physical quantities:

b_ℓ	=	volume factor for phase ℓ
c_ℓ	=	phase compressibility
c_t	=	total compressibility
f	=	fractional flow
K	=	absolute permeability
p	=	pressure
t	=	time
$\vec{u}$	=	total Darcy velocity
q	=	volumetric rate
λ	=	relative mobility
π	=	face pressure

Vectors and matrices:

$\boldsymbol{p}$	=	vector of cell/block pressures
$\boldsymbol{\pi}$	=	vector of cell/block face pressures
$\boldsymbol{u}$	=	vector of outward fluxes on cell/block faces
$\boldsymbol{B}$	=	inner product of velocity basis functions
$\boldsymbol{C}$	=	integral of the divergence of velocity b.f.
$\boldsymbol{D}$	=	map from local to global face numbering
$\boldsymbol{P}$	=	compressibility matrix
$\boldsymbol{T}_E$	=	transmissibility matrix for cell E
$\boldsymbol{V}$	=	integral of quadratic velocity terms
$\boldsymbol{\Psi}$	=	matrix of all basis functions

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 $\vec{\psi}_{ij}$

Basis functions, etc:

$\vec{\psi}_{ij}$	=	basis function, interface of block i and j
w_i	=	weight function associated with coarse block Ω_i

Numbers:

n_E	=	number of faces in cell E

Subscripts:

i, j	=	block/cell numbers
ℓ	=	Phase number (o=oil, w=water, g=gas)
ν	=	iteration number
f	=	vector/matrix defined relative to the fine grid

Superscripts:

n	=	time step

References

Aarnes, J. E., Kippe, V., and Lie, K.-A. 2006a. Multiscale Methods and Streamline Simulation for Rapid Reservoir Performance Prediction. In *Progress in Industrial Mathematics at ECMI 2004*, volume 8 of *Math. Ind.*, pages 399–403. Springer, Berlin.

Aarnes, J. E., Krogstad, S., and Lie, K.-A. 2006b. 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.

Bockel-Rebelle, M. O., Dabbour, Y., El Abd Salem, S., Vesseron, M., and Silva, F. P. 2005. Fault and Fracture Corridors–How to Reduce the Structural Uncertainty for Reservoir Management Optimization. Paper SPE 93752 presented at 14th SPE Middle East Oil & Gas Show and Conference, Bahrain, March 12–15.

Brezzi, F., Lipnikov, K., and Shashkov, M. 2005a.Convergence of the Mimetic Finite Difference Method for Diffusion Problems on Polyhedral Meshes.*SIAM J. Numer Anal.*, 43(5):1872–1896.

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

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

Cheng, H., Wen, X.-H., Milliken, W. J., and Datta-Gupta, A. 2004. Field Experiences with Assisted and Automatic History Matching Using Streamline Models. Paper SPE 89857 presented at 2004 SPE Annual Technical Conference and Exhibition, Houston, Texas, 26–29 September.

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/`.

Grinestaff, G. H. and Caffrey, D. J. 2000. Waterflood Management: A Case Study of the Northwest Fault Block Area of Prudhoe Bay, Alaska, Using Streamline Simulation and Traditional Waterflood Analysis. Paper SPE 63152 presented at 2000 SPE Annual Technical Conference and Exhibition, Dallas, Texas, 1–4 October.

Hajibeygi, H., Bonfigli, G., Hesse, M. A., and Jenny, P. 2008. Iterative Multiscale Finite-Volume Method. *J. Comput. Phys*, 227(19):8604–8621.

Kretz, V., Valles, B., and Sonneland, L. 2004. Fluid Front History Matching Using 4D Seismic and Streamline Simulation. Paper SPE 90136 presented at 2004 SPE Annual Technical Conference and Exhibition, Houston, Texas, 26–29 September.

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 SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February.

Lolomari, T., Bratvedt, K., Crane, M., Milliken, W. J., and Thyrie, J. J. 2000. The Use of Streamline Simulation in Reservoir Management: Methodology and Case Studies. Paper SPE 63156 presented at 2000 SPE Annual Technical Conference and Exhibition, Dallas, Texas, 1–4 October.

Matthäi, S. K., Mezentsev, A., and Belayneh, M. 2007. Finite Element, Node-Centered Finite-Volume Two-Phase-Flow Experiments With Fractured Rock Represented by Unstructured Hybrid-Element Meshes. *SPE Reserv. Eval. Eng.*, 10(6):740–756. SPE 93341.

Montaron, B., Bradley, D., Cooke, A., Prouvost, L., Raffn, A. G., Vidal, A., and Wilt, M. 2007. Shapes of Flood Fronts in Heterogeneous Reservoirs and Oil Recovery Strategies. Paper SPE 111147 presented at SPE/EAGE Reservoir Characterization and Simulation Conference, Abu Dhabi, UAE, October 28–31.

Moreno, J., Kazemi, H., and Gilman, J. 2004. Streamline Simulation of Countercurrent Water-Oil and Gas-Oil Flow in Naturally Fractured Dual-Porosity Reservoirs. Paper SPE 89880 presented at 2004 SPE Annual Technical Conference and Exhibition, Houston, Texas, 26–29 September.

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.

Stenerud, V. R., Kippe, V., Lie, K.-A., and Datta-Gupta, A. 2008. Adaptive Multiscale Streamline Simulation and Inversion for High-Resolution Geomodels. *SPE J.*, 13(1):99–111. SPE 106228-PA.

Stenerud, V. R. Multiscale-Streamline Inversion for High-Resolution Reservoir Models. PhD Thesis, 2007:211, Norwegian University of Science and Technology, Trondheim, Norway.