A Comparison of Multiscale Methods for Elliptic Problems in Porous Media Flow

Vegard Kippe (`vki@sintef.no`)[†], Jørg E. Aarnes (`jaa@sintef.no`)[†]
and Knut-Andreas Lie (`knut-andreas.lie@sintef.no`)[†,‡]
† *SINTEF ICT, Dept. Applied Math., P.O. Box 124 Blindern, N–0314 Oslo, Norway*
‡ *Center of Mathematics for Applications, University of Oslo*

Abstract. We review and perform comparison studies for three recent multiscale methods for solving elliptic problems in porous media flow; the multiscale mixed finite-element method, the numerical subgrid upscaling method, and the multiscale finite-volume method. These methods are based on a hierarchical strategy, where the global flow equations are solved on a coarsened mesh only. However, for each method, the discrete formulation of the partial differential equations on the coarse mesh is designed in a particular fashion to account for the impact of heterogeneous subgrid structures of the porous medium.

The three multiscale methods produce solutions that are mass conservative on the underlying fine mesh. The methods may therefore be viewed as efficient, approximate fine-scale solvers, i.e., as an inexpensive alternative to solving the elliptic problem on the fine mesh. In addition, the methods may be utilized as an alternative to upscaling, since they generate mass-conservative solutions on the coarse mesh. We therefore choose to also compare the multiscale methods with a state-of-the-art upscaling method—the adaptive local-global upscaling method, which may be viewed as a multiscale method when coupled with a mass-conservative downscaling procedure.

We investigate the properties of all four methods through a series of numerical experiments designed to reveal differences with regard to accuracy and robustness. The numerical experiments reveal particular problems with some of the methods, and these will be discussed in detail along with possible solutions. Next, we comment on implementational aspects and perform a simple analysis and comparison of the computational costs associated with each of the methods. Finally, we apply the three multiscale methods to a dynamic two-phase flow case and demonstrate that high efficiency and accurate results can be obtained when the subgrid computations are made part of a preprocessing step and not updated, or updated infrequently, throughout the simulation.

Keywords: porous media flow, multiscale methods, upscaling, numerical comparisons, two-phase simulation

1. Introduction

In recent years, there has been a growing interest in numerical methods specially designed to model multiscale phenomena; that is, phenomena that are governed by physical processes occurring on a wide range of time and/or length scales. These methods are usually referred to as multiscale methods and have the common characteristic that they incorporate fine-scale information into a set of coarse-scale equations in a way that is consistent with the local properties of the mathematical model on the unresolved subscale(s).

In this paper, we consider multiscale methods for the simulation of pressure and (phase) velocities in porous media flow. The primary cause of multiscale behaviour in porous media flow is heterogeneities in rock and sand formations, occurring at all scales from the pore scale to the scale of the entire model. These heterogeneous structures are reflected in the coefficients of the governing partial differential equations. To accurately resolve the pressure distribution, which is modelled by an elliptic (or parabolic) equation, it is generally acknowledged that it is necessary to account for the influence of fine-scale variations in the coefficients to generate reliable solutions. For incompressible flow, the pressure equation reduces to the variable-coefficient Poisson equation,

$$-\nabla(\mathbf{a}(\mathbf{x})\nabla p) = q. \tag{1}$$

Here the coefficient $\mathbf{a}(\mathbf{x})$ is a symmetric and positive definite tensor that typically has a multiscale structure due to the strongly heterogeneous nature of natural porous media. By multiscale structure we mean that the values of $\mathbf{a}$ span several orders of magnitude, or alternatively that the length scale of variation depends strongly on the position. If $\mathbf{a}$ is drawn from a stochastic distribution, a multiscale structure means that the correlation lengths in the stochastic model vary significantly in space.

The literature on numerical methods for elliptic problems contains a number of multiscale methods that are geared toward solving problems with highly oscillatory coefficients. Examples include the multiscale finite-element method [21], the variational multiscale method [22], residual free bubbles [9, 31], the two-scale finite-element method [28], the two-scale conservative subgrid approach [5, 6], the mixed multiscale finite-element method [13, 1], and the multiscale finite-volume method [25, 24]. All of these methods are based on a hierarchical two-scale approach, where the general idea is to derive a set of equations on a coarse scale that embody the impact of subgrid variations in the elliptic coefficients. To this end, subgrid computations are performed as part

of the multiscale method to estimate how these fine-scale variations influence the coarse-grid solution.

Multiscale methods for elliptic problems offer large computational savings when applied as direct solvers for problems with scale separation. For problems without scale separation—like pressure equations encountered in porous media flows, which generally do *not* have scale separation—the computational complexity of a full solution by a multiscale method is about the same as the complexity of solving the same problem discretized on the subgrid scale with a very efficient linear solver. However, multiscale methods still offer a potential for large computational savings. First of all, in most models of porous media flow, the pressure equation is time-dependent—either by itself or as part of a larger model, as in multiphase flow—and therefore has to be solved repeatedly. Because the temporal changes in $\mathbf{a}(\mathbf{x})$ are typically moderate compared to the spatial variability, it is seldom necessary to perform the subgrid computations every time the pressure equation is solved. Hence, it is possible to obtain an approximate solution on a fine scale at the cost of solving the same problem on a coarsened mesh, leading to significant savings in computational time. Secondly, the subgrid computations in multiscale methods are usually defined in such a way that they can be easily parallelized or used as a kind of domain-decomposition method to solve problems that would otherwise not fit in memory. In other words, it can be beneficial to utilize multiscale methods, also for problems without scale separation.

The purpose of this paper is to compare and validate three selected multiscale methods in terms of accuracy, robustness, and computational complexity; a mixed multiscale finite-element method [13, 1], a two-scale conservative subgrid approach [5, 6], and a multiscale finite-volume method [25, 24]. In addition, we consider a state-of-the-art upscaling method, the adaptive local-global method [14]. This method has similarities with multiscale methods, and can be viewed as an upscaling–downscaling analogue to the multiscale methods when supplied with a procedure for reconstructing a global subgrid solution from the coarse-scale solution. Each method will be applied to solve the pressure equation in immiscible and incompressible two-phase flow models. The numerical examples range from test cases with realistic heterogeneous structures to synthetic cases that are specially designed to reveal limitations in the methods.

The paper is organized as follows: The governing equations for immiscible and incompressible two-phase flow are described in Section 2. The multiscale mixed finite-element method and the two-scale conservative subgrid approach are described in Section 3. In Section 4, we describe the multiscale finite-volume method and the adaptive local-

global method. The numerical experiments, which form the main part of the paper, are performed and discussed in Section 5. In Section 6 we discuss computational aspects of the methods and give a simple analysis of their computational efficiency. Finally, the multiscale methods are applied to a dynamic two-phase flow problem in Section 7, before the paper is summarized and some conclusions are given in Section 8.

2. Governing equations

Immiscible and incompressible two-phase flow in porous media can be described by a coupled system of partial differential equations consisting of an elliptic equation for fluid pressure and a transport equation for the movement of fluid phases. For simplicity of presentation, we will henceforth neglect effects from gravity and capillary forces, and assume that the porosity is constant. If we denote the two phases by o and w (for oil and water), we obtain the following variable-coefficient Poisson equation for pressure p and total Darcy velocity $\mathbf{u}$,

$$\nabla \cdot \mathbf{u} = q, \quad \mathbf{u} = -\mathbf{K}\lambda_t \nabla p \quad \text{in } \Omega, \tag{2}$$

and a hyperbolic equation for the water saturation S,

$$\frac{\partial S(x,t)}{\partial t} + \nabla \cdot (f_w(S)\mathbf{u}) = \max\{q,0\} + f_w(S)\min\{q,0\}. \tag{3}$$

The total velocity $\mathbf{u} = \mathbf{u}_o + \mathbf{u}_w$ is a sum of the phase velocities, q is a volumetric source term representing wells, and the rock permeability tensor $\mathbf{K}$ is assumed to be symmetric and uniformly positive definite. The total mobility is given by $\lambda_t = \lambda_w + \lambda_o$, where λ_i models the reduced mobility of phase i due to the presence of the other phase. Finally, $f_w = \lambda_w/\lambda_t$ denotes the fractional flow of water.

Throughout the paper we assume homogeneous initial conditions and impose homogeneous Neumann boundary conditions:

$$S(\cdot,0) = 0 \text{ in } \Omega \qquad \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega.$$

Here $\mathbf{n}$ is the outward-pointing unit normal on $\partial\Omega$. Furthermore, for simplicity we assume that $\lambda_w = S$ and $\lambda_o = 1 - S$, hence $\lambda_t \equiv 1$, and the pressure and saturation equations are effectively decoupled. In Section 7 we change this assumption and consider a dynamic two-phase flow, where there is a coupling between the pressure and saturation. In all the following numerical simulations, the saturation equation (3) is solved using an explicit single-point upwind finite-volume method.

To discretise (2) using one of the multiscale methods described in this paper, one must first introduce a fine and a coarse mesh. The fine

mesh is usually a nested subgrid of the coarse mesh, and is therefore often referred to as the subgrid. In this paper we limit our attention to Cartesian meshes only. Hence, we assume that Ω has been partitioned into a conforming fine Cartesian mesh $\mathcal{T}_h$, and that a coarse Cartesian mesh $\mathcal{T}_H$ is formed as a collection of elements in $\mathcal{T}_h$. In addition, for implementational simplicity we assume that $\mathbf{K}$ is a constant diagonal tensor in each cell $E \in \mathcal{T}_h$. This implies that $\mathcal{T}_h$ is so-called $\mathbf{K}$-orthogonal, i.e., that $\mathbf{n}_i \cdot \mathbf{K}(E)\mathbf{n}_j = 0$ where n_i and n_j are unit vectors normal to faces perpendicular to coordinate directions i and j $(i \neq j)$, respectively.

There are several reasons for restricting our attention to Cartesian meshes. First, not all the multiscale methods described below are applicable to general meshes. All methods may in principle be extended to more complex meshes, but implementing extensions is a non-trivial task for some of the methods, as will be briefly discussed in Section 6. In fact, to our knowledge, only the multiscale mixed finite-element method has been applied to non-Cartesian meshes [3, 2]. Hence, performing comparisons of the multiscale methods on non-Cartesian meshes would favour the multiscale mixed finite-element method, which is more flexible with respect to meshes, and therefore make it hard to make fair conclusions concerning the main objective in this paper—to assess the ability of the respective multiscale methods to correctly capture the influence of subgrid heterogeneous structures.

3. Description of the Finite-Element Based Methods

Both the multiscale mixed finite-element method and the two-scale conservative subgrid approach, henceforth called the numerical subgrid upscaling method, are based on a mixed formulation. For our model problem (2), with the aforementioned assumption $\lambda_t \equiv 1$, the mixed formulation reads: Find $(\mathbf{u},\, p) \in H_0^{\mathrm{div}}(\Omega) \times L^2(\Omega)$ such that,

$$
\begin{aligned}
(\mathbf{K}^{-1}\mathbf{u},\, \mathbf{v}) \; - \; (p, \nabla \cdot \mathbf{v}) \; &= \; 0, & \forall \mathbf{v} \in H_0^{\mathrm{div}}(\Omega), \\
(\nabla \cdot \mathbf{u},\, l) \; &= \; (q, l), & \forall l \in L^2(\Omega).
\end{aligned}
\tag{4}
$$

Here $H_0^{\mathrm{div}}(\Omega) = \{\mathbf{v} \in L^2(\Omega)^n \; : \; \nabla \cdot \mathbf{v} \in L^2(\Omega),\; \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\}$, and $L^2(\Omega)$ is the space of square integrable functions over Ω.

3.1. Standard Mixed Finite-Element Methods

In mixed finite-element methods for (2) (with $\lambda_t \equiv 1$) one seeks an approximate solution $(\mathbf{u}_h, p_h)$ of (4) that is confined to lie in finite-dimensional subspaces $\mathbf{V} \subset H_0^{\mathrm{div}}(\Omega)$ and $W \subset L^2(\Omega)$. The discrete

formulation reads: find $(\mathbf{u}_h,\ p_h) \in \mathbf{V} \times W$ such that,

$$
\begin{aligned}
(\mathbf{K}^{-1}\mathbf{u}_h,\ \mathbf{v}_h) - (p_h,\ \nabla \cdot \mathbf{v}_h) &= 0, & \forall \mathbf{v}_h \in \mathbf{V}, \\
(\nabla \cdot \mathbf{u}_h,\ l_h) &= (q,\ l_h), & \forall l_h \in W.
\end{aligned}
\tag{5}
$$

Below we make use of two standard mixed finite-element methods, the lowest-order Raviart-Thomas method (RT0) [29] and the lowest-order Brezzi–Douglas–Marini method (BDM1) [11]. In both methods, W equals the space $\mathcal{P}_0(\mathcal{T})$ of piecewise constants over the given mesh $\mathcal{T}$ and the methods only differ in how $\mathbf{V}$ is defined. For RT0,

$$
\mathbf{V} \subset \{\mathbf{v} \subset H_0^{\mathrm{div}}(\Omega) \ : \ \nabla \cdot \mathbf{v} \in W,\ \mathbf{v} \cdot \mathbf{n} \in \mathcal{P}_0(\Gamma)\},
$$

where $\Gamma = \{\partial T_i \cap \partial T_j : T_i, T_j \in \mathcal{T}\}$ and $\mathbf{n}$ is a uniquely oriented unit normal. For BDM1,

$$
\mathbf{V} \subset \{\mathbf{v} \subset H_0^{\mathrm{div}}(\Omega) \ : \ \nabla \cdot \mathbf{v} \in W,\ \mathbf{v} \cdot \mathbf{n} \in \mathcal{P}_1(\Gamma)\}.
$$

Thus, whereas the RT0 space restricts the normal component of the velocity on each interface to be constant, the normal component of the velocity on each interface in BDM1 is allowed to be linear.

The velocity approximation spaces in mixed methods are usually spanned by basis functions associated with interfaces in the mesh. For RT0 there is only one degree of freedom per interface. Hence, only one basis function per interface is needed. The RT0 basis functions may therefore be thought of as representing flow units across element interfaces. In BDM1, the velocity space contains, in addition to the RT0 basis functions, a set of "divergence-free" basis functions. These basis functions are divergence free in the sense that their divergence is orthogonal to W with respect to the L^2-norm. An explicit listing of these basis functions can be found in [6] and details of the mathematical properties of the RT0 and BDM1 methods can be found in, e.g., [10].

3.2. Multiscale Mixed Finite-Element Method (MsMFEM)

The multiscale method we consider here belongs to a family of multiscale finite-element methods, first introduced by Hou and Wu [21]. The basic idea of the methods is to construct special basis functions that are adaptive to the local properties of the elliptic differential operator. To ensure local mass conservation, Chen and Hou [13] introduced a multiscale method based on a mixed finite-element discretization. This method gives mass-conservative velocity fields on the coarse mesh and also on the fine mesh in coarse blocks not containing sources. In the following, we employ a similar mixed finite-element method due to Aarnes and Lie [1, 4, 2] that provides mass-conservative velocity fields on the entire fine mesh.

On the coarse scale, MsMFEM is a generalization of the standard RT0 method, searching for a solution in the discrete approximation spaces $(\mathbf{V}^{\mathrm{ms}}, \mathcal{P}_0(\mathcal{T}_H))$, where the piecewise linear velocity basis functions of the RT0 method on $\mathcal{T}_H$ are generalized to account for subgrid variations in the coefficients. This is accomplished by letting the basis functions be solutions of (2) restricted to a pair of coarse-mesh elements with source terms specified in such a way that unit flow is forced across the element interface. Specifically, if E_i and E_j denote two coarse-mesh elements with a common interface $\Gamma_{ij} = \partial E_i \cap \partial E_j$, the multiscale velocity basis functions $\boldsymbol{\psi}_{ij}$ are defined as follows,

$$
\begin{aligned}
\boldsymbol{\psi}_{ij} &= -\mathbf{K}\nabla\phi_{ij}, \\
\nabla \cdot \boldsymbol{\psi}_{ij} &= \begin{cases} w_i(x)/\int_{E_i} w_i(\xi)\, \mathrm{d}\xi, & \text{for } x \in E_i, \\ -w_j(x)/\int_{E_j} w_j(\xi)\, \mathrm{d}\xi, & \text{for } x \in E_j, \end{cases} \\
\boldsymbol{\psi}_{ij} \cdot \mathbf{n} &= 0, \qquad \text{on } \partial(E_i \cup \Gamma_{ij} \cup E_j).
\end{aligned}
\tag{6}
$$

For certain element types, e.g., triangles and tetrahedrons, or rectangular parallelepipeds with a diagonal permeability tensor, these basis functions reduce to the standard RT0 basis functions if $\mathbf{K}$ and w_i are constants. We note that the definition (6), first introduced in [4], is slightly different from the definitions in [13] and [1]. The advantage of solving simultaneously in both E_i and E_j is that we avoid explicit specification of boundary conditions on Γ_{ij}.

In [1] it was noted that the MsMFEM solution will be locally mass conservative if the local source term w_i coincides with q in elements containing sources or sinks. For elements where $q = 0$, we may choose w_i arbitrarily, but the properties of the basis functions will depend on the particular choice. This issue was investigated in [2], and here we follow the recommendation of scaling w_i according to the trace of the permeability tensor; i.e., we use

$$
w_i(x) = \begin{cases} \mathrm{trace}(\mathbf{K}(x)), & \text{if } q(x)|_{E_i} = 0, \\ q(x), & \text{otherwise.} \end{cases}
\tag{7}
$$

This completes the definition of the MsMFEM velocity basis functions $\boldsymbol{\psi}_{ij}$, and the approximation space $\mathbf{V}^{\mathrm{ms}}$ is given as $\mathrm{span}\{\boldsymbol{\psi}_{ij}\}$.

3.3. The Numerical Subgrid Upscaling Method (NSUM)

Instead of generalizing the coarse-scale basis functions, as in MsM-FEM, the approach in NSUM is to enrich the coarse-scale approximation space with a family of mutually orthogonal approximation spaces that contain local fine-scale variations. Although NSUM was originally formulated as an RT0 method on the coarse scale [8], we

here follow [5, 6] and let the coarse-scale approximation space be the BDM1 space. Subgrid variations in the solution will be modelled using RT0 approximation spaces.

Let E_c be a coarse element in $\mathcal{T}_H$ and denote by $W_h(E_c)$ the fine-scale RT0 pressure space restricted to E_c, with the additional constraint of zero average:

$$W_h(E_c) = \{w_h \in \mathcal{P}_0(\mathcal{T}_h)|_{E_c} : (w_h,\, 1)_{E_c} = 0\}.$$

Furthermore, let $\mathbf{V}_h(E_c)$ be the RT0 velocity space with support strictly inside E_c; i.e.,

$$\mathbf{V}_h(E_c) = \{\mathbf{v}_h \in \mathbf{V}^{\mathrm{RT0}}(\mathcal{T}_h)|_{E_c} : \mathbf{v}_h \cdot \mathbf{n} = 0 \text{ on } \partial E_c\}.$$

Finally, let $(\mathbf{V}_H,\, W_H)$ be the BDM1 approximation spaces over the coarse mesh $\mathcal{T}_H$. The NSUM approximation spaces are then given as the following direct sums,

$$
\begin{aligned}
W_{H,h} &= W_H \bigoplus_{E_c \in \mathcal{T}_H(\Omega)} W_h(E_c) = W_H \oplus W_h, \\
\mathbf{V}_{H,h} &= \mathbf{V}_H \bigoplus_{E_c \in \mathcal{T}_H(\Omega)} \mathbf{V}_h(E_c) = \mathbf{V}_H \oplus \mathbf{V}_h.
\end{aligned}
\tag{8}
$$

Each $(\mathbf{u},\, p) \in (\mathbf{V}_{H,h},\, W_{H,h})$ may then be uniquely decomposed into $\mathbf{u} = \mathbf{u}_H + \mathbf{u}_h$ and $p = p_H + p_h$ with $(\mathbf{u}_H,\, p_H) \in (\mathbf{V_H},\, W_H)$ and $(\mathbf{u}_h,\, p_h) \in (\mathbf{V}_h,\, W_h)$. Substituting these representations into (4) and testing with subgrid test-functions yields a set of equations for each coarse element,

$$
\begin{aligned}
(\mathbf{K}^{-1}\mathbf{u}_h,\, \mathbf{v}_h)_{E_c} - (p_h,\, \nabla \cdot \mathbf{v}_h)_{E_c} &= -(\mathbf{K}^{-1}\mathbf{u}_H,\, \mathbf{v}_h)_{E_c}, \\
(\nabla \cdot \mathbf{u}_h,\, l_h)_{E_c} \qquad\qquad &= (q - \nabla \cdot \mathbf{u}_H,\, l_h)_{E_c},
\end{aligned}
\tag{9}
$$

for all $\mathbf{v}_h \in \mathbf{V}_h(E_c)$ and all $l_h \in W_h(E_c)$. Testing with coarse-scale test-functions in (4) yields a global set of equations,

$$
\begin{aligned}
(\mathbf{K}^{-1}(\mathbf{u}_H + \mathbf{u}_h),\, \mathbf{v}_H) - (p_H,\, \nabla \cdot \mathbf{v}_H) &= 0, \\
(\nabla \cdot \mathbf{u}_H,\, l_H) &= (q,\, l_H),
\end{aligned}
\tag{10}
$$

for all $\mathbf{v}_H \in \mathbf{V}_H$ and all $l_H \in W_H$.

The subgrid equations (9) are mutually decoupled, but they are all coupled to the coarse-scale solution $\mathbf{u}_H$, and hence to the coarse-scale equation (10). However, a solution $\mathbf{u}_h$ of (9) may be expressed in terms of coarse-scale basis coefficients and the computed actions of the subgrid operator on the coarse-scale basis functions. Let us express the coarse-scale solution using a local numbering of basis functions,

$$\mathbf{u}_H = \sum_{E_c \in \mathcal{T}_H} \sum_{j>0} \beta^{E_c,j} \mathbf{v}_H^{E_c,j}. \tag{11}$$

For each coarse element E_c, it is easy to verify that the solution to the subgrid equations (9) may be written on the following form:

$$\mathbf{u}_h = \mathbf{u}_h^0 + \sum_{j>0} \beta^j \mathbf{u}_h^j, \qquad p_h = p_h^0 + \sum_{j>0} \beta^j p_h^j, \qquad (12)$$

where $(\mathbf{u}_h^0, p_h^0)$ solves

$$
\begin{aligned}
(\mathbf{K}^{-1}\mathbf{u}_h^0, \mathbf{v}_h)_{E_c} - (p_h^0, \nabla \cdot \mathbf{v}_h)_{E_c} &= 0, \\
(\nabla \cdot \mathbf{u}_h^0, l_h)_{E_c} &= (q - \mathcal{P}_{W_H} q, l_h)_{E_c},
\end{aligned}
\qquad (13)
$$

for all $\mathbf{v}_h \in \mathbf{V}_h(E_c)$ and all $l_h \in W_h(E_c)$, and $(\mathbf{u}_h^j, p_h^j)$ for $j > 0$ solves

$$
\begin{aligned}
(\mathbf{K}^{-1}\mathbf{u}_h^j, \mathbf{v}_h)_{E_c} - (p_h^j, \nabla \cdot \mathbf{v}_h)_{E_c} &= -(\mathbf{K}^{-1}\mathbf{v}_H^j, \mathbf{v}_h)_{E_c}, \\
(\nabla \cdot \mathbf{u}_h^j, l_h)_{E_c} &= 0,
\end{aligned}
\qquad (14)
$$

for all $\mathbf{v}_h \in \mathbf{V}_h(E_c)$ and all $l_h \in W_h(E_c)$. Here $\mathcal{P}_{W_H}$ represents a projection onto W_H, and we have $\mathcal{P}_{W_H} q = \nabla \cdot \mathbf{u}_H$ by (10) and the fact that the BDM1 approximation spaces $(W_H, \mathbf{V}_H)$ satisfy $\nabla \cdot \mathbf{V}_H = W_H$. A discrete system is now obtained by inserting (12) into (10), and possibly adding some subgrid equations to symmetrize the system [6].

Finally, we note that the NSUM formulation allows virtually any mixed finite-element method on both the coarse and fine scales [7], but, to our knowledge, only the RT0 and BDM1 methods have so far been used in practice.

4. Description of the Finite-Volume Based Methods

The two remaining methods, the multiscale finite-volume method and the adaptive local-global upscaling method, are both based on a finite-volume formulation. In a finite-volume method for (2) (with $\lambda_t = 1$) one introduces a family of control volumes (typically the given mesh) and imposes mass conservation locally for each control volume E:

$$\int_{\partial E} -\mathbf{K}\nabla p \cdot \mathbf{n}\, ds = \int_E q\, dx, \qquad (15)$$

where $\mathbf{n}$ is the outward unit normal on ∂E. The simplest and most widely used finite-volume method is the two-point flux approximation (TPFA) scheme. The TPFA scheme is a cell-centred finite-volume method, which may be expressed as a pressure stencil for each cell E_i,

$$\sum_j T_{ij}(p_i - p_j) = q_i, \qquad (16)$$

where j loops over all cells neighbouring cell E_i. The transmissibilities T_{ij} are associated with cell interfaces, and the expression $T_{ij}(p_i - p_j)$ is a discrete form of

$$-\int_{\partial E_i \cap \partial E_j} \mathbf{K}\nabla p \cdot \mathbf{n}_{ij}\, ds, \tag{17}$$

where $\mathbf{n}_{ij}$ is the unit normal on $\Gamma_{ij} = \partial E_i \cap \partial E_j$ pointing into E_j. Thus, $T_{ij}(p_i - p_j)$ estimates the total flux across $\partial E_i \cap \partial E_j$.

4.1. The Multiscale Finite-Volume Method (MsFVM)

This method was introduced by Jenny et al. [25, 24] and is a control-volume finite-element formulation on the coarse mesh $\mathcal{T}_H$. Here the pressure p is expressed as a linear combination of basis functions ϕ_i that incorporate subscale pressure variations (see below). Thus, for each control-volume $E_i \subset \mathcal{T}_H$, we have the following equation:

$$
\begin{aligned}
-\int_{\partial E_i} \mathbf{n} \cdot \mathbf{K}\nabla \left(\sum_j p_j \phi_j\right) ds &= \\
-\sum_j p_j \int_{\partial E_i} \mathbf{n} \cdot \mathbf{K}\nabla \phi_j \, ds &= \\
\sum_j p_j f_{ij} &= \int_{E_i} q \, dx.
\end{aligned}
\tag{18}
$$

The quantity f_{ij} denotes the flux over the boundary of cell E_i due to the basis function centred in cell E_j, and we refer to these quantities as the MsFVM transmissibilities. Equations (18) give a linear system that can be solved for $\{p_j\}$, and a fine-scale pressure solution is given as a linear superposition of the basis functions, i.e., $p = \sum_j p_j \phi_j$.

Since the coarse elements are used as control volumes, the fine-scale pressure solution gives a velocity field that is mass conservative on the coarse mesh $\mathcal{T}_H$. However, this velocity field is generally not mass conservative on the fine mesh $\mathcal{T}_h$. A reconstruction step is therefore necessary to get a mass-conservative velocity solution on the fine scale. This can be achieved by solving (2) within each control volume using Neumann boundary conditions given by the non-conservative velocity solution. That is, for each $E \in \mathcal{T}_H$ one defines the velocity $\mathbf{v}$ inside E by

$$\mathbf{v} = -\mathbf{K}\nabla u, \quad \nabla \cdot \mathbf{v} = q \quad \text{in } E, \quad \mathbf{v} \cdot \mathbf{n} = -\mathbf{K}\nabla p \cdot \mathbf{n} \text{ on } \partial E, \tag{19}$$

where $p = \sum_j p_j \phi_j$ is the MsFVM pressure solution.

An alternative, but equivalent reconstruction procedure was described in [25, 24]. Here the idea is to express the reconstructed velocity

field as a linear superposition of basis functions analogous to the way the fine-scale velocity field is obtained in MsMFEM. To this end, for each pressure basis function ϕ_j, one associates a reconstruction basis function with each cell in the coarse mesh $\mathcal{T}_H$ where ϕ_j is supported. Although the number of reconstruction basis functions may be large—twenty-seven for each basis function ϕ_j on a Cartesian mesh in three dimensions—the latter reconstruction procedure may be more computational efficient than (19) for two-phase flow simulation when only a few basis functions and/or transmissibilities need to be recomputed [24]. For instance, for Cartesian meshes in three dimensions, computational savings are obtained if less than four percent of the basis functions and/or transmissibilities are recomputed. It should be noted, however, that the reconstruction procedure in (19) allows inclusion of additional physical effects, such as gravity and compressibility [27], and that performing the reconstruction everywhere, as is done in (19), in our experience makes the MsFVM more robust. We will return to this issue in Section 7.

Pressure Basis Functions
To describe the computation of the MsFVM basis functions, we first introduce a dual mesh $\overline{\mathcal{T}}_H$ by connecting the cell centres of adjacent cells. The cell centres of the original mesh then become the nodes of the dual mesh, and vice versa, see Figure 1. Each pressure basis function for MsFVM is now associated with a coarse element and is supported in all cells of the dual mesh that have the centre of the associated coarse cell as a common vertex. The left plot in Figure 1 shows the support of a basis function along with parts of the coarse and fine meshes. Also shown is parts of the dual coarse mesh. To construct a basis function, one solves a homogeneous version of (2) within all cells of the dual coarse mesh that share a common vertex. For instance, for the case depicted in Figure 1, we solve one subgrid problem for each of the shaded cells in the dual coarse mesh. Next we describe how to construct the part of the basis function ϕ associated with $x_{i,j}$ inside the dark-shaded cell in the dual mesh. The other parts of ϕ are defined similarly.

The right plot in Figure 1 shows the lower-left (dark-shaded) dual cell from the left plot. To localize the basis function ϕ to the shaded area, the boundary condition along Γ_1 and Γ_3 must be zero. Next we require that $\phi = 1$ at the internal node $x_{i,j}$. Then it only remains to define boundary conditions on Γ_2 and Γ_4. These boundary conditions should ideally reflect the flow behaviour through the dual cell when embedded in the global system. In [25], the authors follow the idea from the multiscale finite-element method [21] and propose to solve
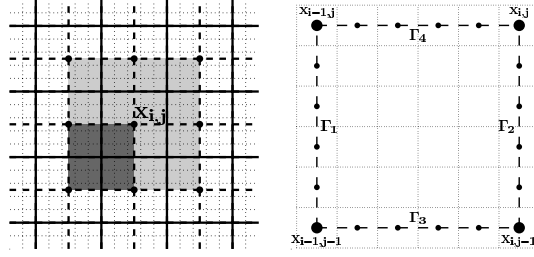
Figure 1. Domain of the basis functions associated with node $x_{i,j}$ (shaded) along with the fine mesh (dotted), the coarse mesh (thick solid), and the dual mesh (thick dashed). To the right is the highlighted dual cell in more detail.

a reduced problem along the boundary to determine boundary conditions. Denoting the dark-shaded dual cell by $\overline{E}$, the equations that define the pressure basis function ϕ read,

$$-\nabla \cdot \mathbf{K}\nabla\phi = 0 \text{ in } \overline{E}, \qquad \phi = b \text{ on } \partial\overline{E}, \qquad (20)$$

with the boundary condition b given by,

$$-\nabla \cdot \mathbf{K}_i \nabla b = 0 \text{ on } \Gamma_i \qquad b(x_{i,j}) = 1, \qquad b(x_{i,k}) = 0 \text{ for } k \neq j. \quad (21)$$

Here $\mathbf{K}_i = \mathbf{r}_i \cdot \mathbf{K}\mathbf{r}_i$, where $\mathbf{r}_i$ is the unit vector tangential to Γ_i.

4.2. Adaptive Local-Global Upscaling Coupled With Nested Gridding (ALGUNG)

This method was introduced by Chen et al. [15, 14] and is an iterative method for computing upscaled permeabilities or transmissibilities that account for global flow effects as well as the impact of small-scale features in an averaged sense. The idea is to use a global coarse-scale pressure solution $p^{c,n-1}$ to determine boundary conditions for an extended local computation of upscaled quantities. Using these quantities, a new coarse-scale solution $p^{c,n}$ may be computed, and the process is repeated until the computed quantities are self-consistent. Once a satisfactory coarse-scale solution has been found, we may reconstruct a conservative fine-scale velocity field using a suitable reconstruction procedure. Here we shall employ the reconstruction procedure from the nested-gridding method [20], and, following [15, 14], we iterate until

$$\frac{\|p^{c,n} - p^{c,n-1}\|_2}{\|p^{c,1} - p^{c,0}\|_2} < 0.01. \qquad (22)$$

The adaptive method introduced in [14] differs from the original local-global approach [15] in two respects. The adaptive version per-
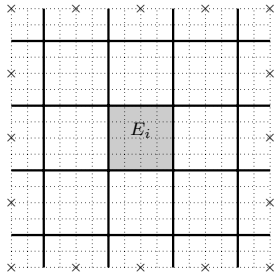
Figure 2. Domain $\mathcal{D}_i$ ($r = 1.5$) for computing transmissibilities associated with the interfaces of coarse cell E_i. The crosses mark the locations of the coarse-scale pressures that are interpolated linearly to give the boundary conditions.

forms the upscaling for a specific flow scenario instead of generic axes-oriented flows, and also applies a thresholding scheme such that up-scaled quantities are only recomputed in high-flow regions. The thresh-olding is motivated both by computational efficiency and by a reduced number of anomalous upscaled values, which typically appear in low-flow regions. In our experience, the thresholding has a slight negative impact on the accuracy of the method, but the loss of accuracy is typi-cally small compared to the reduction in computational work. However, to be certain not to underestimate the performance of the method, we have chosen to disregard the thresholding and recompute all quantities in every iteration when assessing the accuracy of the method.

Chen et al. [15] demonstrated that transmissibility upscaling typ-ically gives better results than permeability upscaling. On the coarse scale, we therefore use upscaled transmissibilities T_{ij}^c rather than up-scaled permeabilities. To simplify the implementation and improve the efficiency, we use a slightly modified version of the algorithm from [15], in which we solve a local fine-scale problem for each coarse block, in-stead of a fine-scale problem for each interface as in [15]. This generally reduces the number of fine-scale problems that we have to solve by a factor equal the number of space dimensions. On the other hand, we generally obtain two transmissibilities for each interface Γ_{ij}. We therefore need a procedure for deriving unique transmissibilities. If both transmissibilities are positive, the average is used. If one of the transmissibility values are negative, the positive value is used. If both transmissibilities are negative, the result from the previous step is used.

To compute the upscaled transmissibilities associated with the in-terfaces of coarse cell E_i, we define a corresponding extended domain $\mathcal{D}_i$. In Figure 2 an extended domain is obtained by connecting centres in a ring of coarse cells surrounding E_i. This approach may give un-necessarily large domains and make ALGUNG unnecessarily expensive, especially in three dimensions. Wen et al. [33] therefore proposed an

alternative approach where the extended domain $\mathcal{D}_i$ is obtained by connecting centres in a ring of fine cells surrounding the local upscaling region.

When the domains for extended local upscaling computations are defined, one needs to introduce an operator that interpolates the coarse-scale pressure p_c onto the boundary of each extended region. In [15, 33] this operator is based on a bilinear or trilinear interpolation between the pressure values associated with cells in the coarse mesh whose centres are closest to the boundary of the extended region. We would like to note that for unstructured grids it may be difficult to decide which coarse cells to use to define a good interpolation operator. Indeed, we believe that generating proper interpolation operators will be a non-trivial task for complex grids, such as corner-point grids with faults and eroded layers that one frequently encounters in reservoir simulation.

After having solved (2) in $\mathcal{D}_i$ with boundary conditions obtained from the previous coarse-scale solution, new upscaled transmissibilities are computed as

$$T_{ij}^c = \frac{q_{ij}^c}{\overline{p}_i - \overline{p}_j}. \tag{23}$$

Here q_{ij}^c is the total flux across the coarse-cell interface Γ_{ij}, and $\overline{p}_i$ and $\overline{p}_j$ are the volume averages of the fine-scale pressure over coarse cells E_i and E_j, respectively. For the initial upscaling step we have used the pressure method as described in, e.g., [18].

Reconstructing the Fine-Scale Solution
We now briefly describe the nested-gridding procedure [20] used to reconstruct a fine-scale solution once the coupled local-global iterations have converged. As for the MsFVM reconstruction, the idea is to solve (2) within each coarse cell subject to Neumann boundary conditions, but here the boundary conditions are scaled according to the fine-scale transmissibilities. Hence, a mass-conservative velocity field on the fine mesh is obtained by solving an elliptic problem of the following form:

$$\begin{aligned} \mathbf{v} = -\mathbf{K}\nabla u, \quad \nabla \cdot \mathbf{v} = q \quad \text{in } E, \\ \mathbf{v} \cdot \mathbf{n}|_\Gamma = q^c \cdot \frac{T_{\gamma_i}}{\sum_{\gamma_j \subset \Gamma} T_{\gamma_j}} \quad \text{on } \Gamma \subset \partial E, \end{aligned} \tag{24}$$

for every $E \in \mathcal{T}_H$. Here q^c is the coarse-scale flux across Γ and T_{γ_i} is the fine-scale transmissibility of interface $\gamma_i \subset \Gamma$.

The only difference between the nested-gridding reconstruction and the reconstruction procedure (19) for MsFVM is the local boundary conditions. Our experience indicates that the nested-gridding method is more robust. This is partially due to the fact that (19) may generate

bi-directional flow across coarse-grid interfaces, whereas the nested-gridding method produces flow only in the same direction as the coarse-scale solution. In Section 5.5, we will show an example where unphysical curls in the MsFVM velocity solution are eliminated if (19) is replaced with the nested-gridding reconstruction.

5. Numerical Experiments

The main objective of this section is to study how well the respective multiscale methods outlined above capture the influence of small-scale heterogeneous structures (small-scale variations in **K**), and to what extent errors in the obtained velocity fields impact the solution of (3) with $f_w(S) = S$. We will begin with a simple example illustrating the typical performance of the methods. Then we assess the robustness of the methods with regard to different coarse meshes, before considering robustness with regard to different heterogeneous structures.

In all the experiments we employ no-flow boundary conditions. The flow conditions are therefore defined by the source and sink terms. Although we will interpret the sources and sinks as injection and production wells, respectively, we will not employ well-models. Instead, we define a well configuration with one injection well with a static injection rate, and a set of production wells with static and equal production rates. This way, we ensure that the total amount of flow through the model is the same for all methods, and hence that errors can be attributed solely to the ability to correctly resolve the impact of small-scale heterogeneous structures. The reason why we have chosen to *not* employ well-models, is that well models are incorporated differently into the respective multiscale methods. Moreover, only limited attempts have been made to design well-models for multiscale methods. Hence, although we acknowledge that well-models may have a strong impact on flow solutions, we feel that well-modelling technology for multiscale methods is not yet mature enough for a comparison study, and that by including well-models we could get ambiguous results.

To study how well the respective multiscale methods model the impact of small-scale heterogeneous structures, we assess the accuracy of the multiscale velocity solutions. To this end, we compute a reference solution $\mathbf{v}^{\mathrm{ref}}$ by applying the TPFA method on a mesh obtained by refining the fine mesh four times in each direction, and project the resulting solution onto the original fine mesh. We then measure velocity solution errors using the following measure:

$$\delta(\mathbf{v}) = \delta(\mathbf{v}_x, \mathbf{v}_y) = \frac{\|\mathbf{v}_x - \mathbf{v}_x^{\mathrm{ref}}\|_2}{\|\mathbf{v}_x^{\mathrm{ref}}\|_2} + \frac{\|\mathbf{v}_y - \mathbf{v}_y^{\mathrm{ref}}\|_2}{\|\mathbf{v}_y^{\mathrm{ref}}\|_2}. \tag{25}$$

Here $\mathbf{v}_x$ and $\mathbf{v}_y$ are vectors containing the average velocities across the fine-mesh interfaces in the x- and y-directions, respectively, and $\|\cdot\|_2$ is the Euclidean norm.

In addition to comparing velocity fields directly using (25), we also want to monitor the impact that errors in the velocity fields has on the solution of the saturation equation (3). In this section we exclude nonlinear effects and use $\lambda_w = S$ and $\lambda_o = 1 - S$. The resulting system (2)–(3) is decoupled, and may be viewed as a model of passive tracer injection with S denoting normalized tracer concentration. We compare saturation fields at $t = 0.3$ PVI using

$$\delta(S) = \frac{\|S - S^{\mathrm{ref}}\|_2}{\|S^{\mathrm{ref}}\|_2}, \tag{26}$$

where the reference solution is computed on the four-times refined mesh. Pore-volumes-injected (PVI) is a standard time-unit in reservoir simulation that refers to the fraction of the total accessible pore volume that has been injected into the domain.

We do not measure errors in the pressure solutions. There are two main reasons for this. First, for incompressible flow simulation, as we consider in this paper, the pressure does not appear explicitly in the saturation equation (3), and therefore pressure influences flow scenarios only implicitly. Secondly, pressure fields are usually much smoother than the associated velocity fields, and, unlike velocity fields, primarily reflect large-scale heterogeneous structures. The pressure is therefore not very well suited to assess the ability to model the influence of small-scale heterogeneous structures correctly.

5.1. 10TH SPE COMPARATIVE SOLUTION PROJECT, MODEL 2

We first consider how the multiscale methods perform on a sequence of two-dimensional models with permeability data from Model 2 from the 10th SPE Comparative Solution Project [16], henceforth referred to as the SPE10 model. This model consists of $60 \times 220 \times 85$ cells, each of size 20 ft $\times$ 10 ft $\times$ 2 ft. The top 35 layers represent a prograding near-shore environment, with quite smooth variation in the coefficients. The bottom 50 layers model a fluvial formation with a spaghetti of narrow high-flow channels. Long correlation length structures, such as the high-flow channels in the lower part of the SPE10 model, are generally difficult to model adequately using conventional upscaling methods,

For each layer, we place sources/sinks in a five-spot pattern, i.e., injection in the centre and production in the four corners, and solve (2) using the multiscale methods on a 15×55 coarse mesh. Figure 3 shows

$\delta(\mathbf{v})$
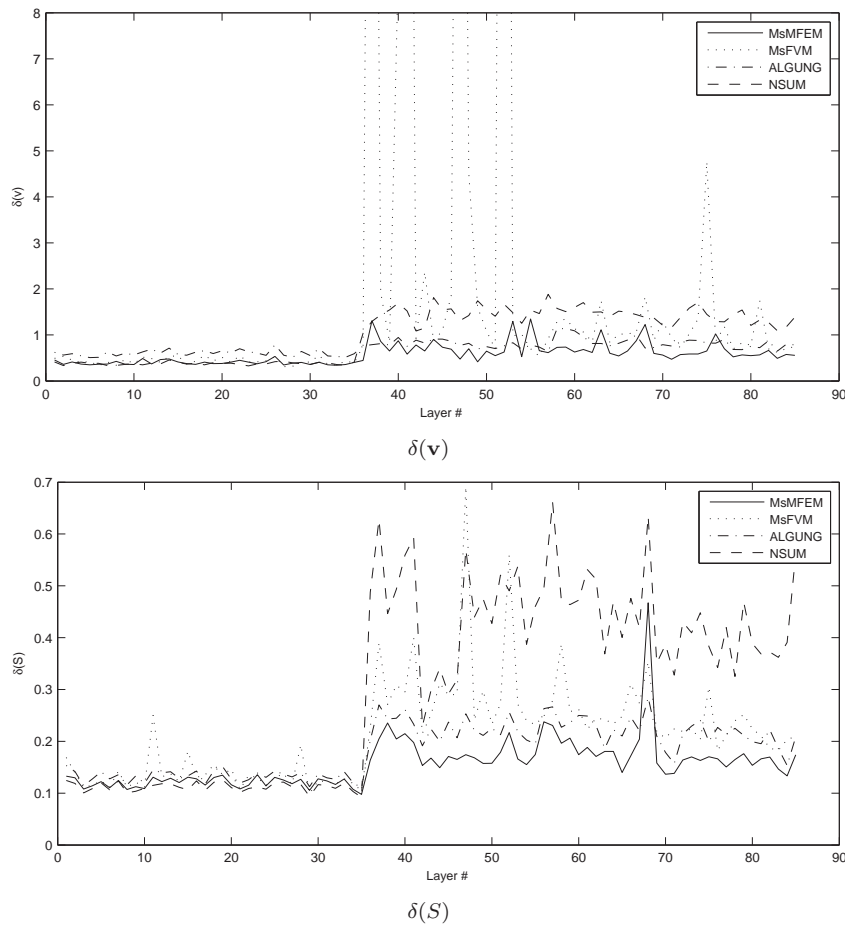

$\delta(S)$

Figure 3. Velocity errors (top) and saturation errors (bottom) for the multiscale methods when using a 15×55 coarse mesh on each layer of the SPE10 model.

velocity and saturation errors for each of the layers. The difference between the smooth layers (1–35) and the fluvial layers (36–85) is striking. In the smooth region, all methods perform reasonably well, with NSUM being the most accurate. For the fluvial region, the situation is reversed, and NSUM is the least accurate method. This is due to the fact that NSUM does not allow non-smooth fine-scale variations on the coarse-mesh interfaces, and is therefore not able to represent the channels accurately. We note that MsFVM yields large velocity errors for a few of the fluvial layers. This problem may occur when there is a large variation in sensitivity to pressure gradients in different coordinate directions, as is the case when high-permeable channels are aligned along one of the axes. We study this problem in detail in

Section 5.5. On average, MsMFEM is the most accurate method for these models, but also ALGUNG generally performs well on all layers. This is reasonable since ALGUNG iterates between a global and local solution in order to capture effects of both small-scale and large-scale structures. The latter is exemplified in the fluvial region, in which the channels represent structures with long correlation lengths.

5.2. Multiscale Methods as Upscaling Methods

Since the multiscale methods are conservative on the fine scale, we may integrate the velocity field and perform simulation on any coarser mesh, thus utilizing the multiscale methods as upscaling methods. In the following we compare saturations obtained using the multiscale methods with saturations obtained using a standard flow-based upscaling method, the numerical pressure computation technique [18], and plain harmonic-arithmetic averaging [12] (see e.g., [30]). The experimental setup is as above, i.e., we solve a five-spot flow problem in each layer of the SPE10 model, but we now try four different coarse meshes. For each coarse mesh, we compute saturation distributions on both the coarse mesh and on the underlying fine mesh and measure errors relative to reference solutions on the corresponding meshes. To generate fine-scale velocities for the local upscaling and simple averaging methods we use the nested-gridding reconstruction, and we refer to the resulting approaches as PUPNG and HANG, respectively.

The average errors over the smooth region (Layers 1–35) are shown in Figure 4, while Figure 5 displays the average errors over the channelized region (Layers 36–85). The fine-scale results demonstrate that the relative performance of the multiscale methods on the different types of data is as seen above, also for other coarse-mesh resolutions. Moreover, we see that for both smooth and channelized data, the two simple upscaling approaches (PUPNG and HANG) perform nearly as well as the multiscale methods when the saturation equation is solved on the coarse mesh, but are outperformed by the multiscale methods when the saturation equation is solved on the fine mesh. Thus, in terms of accuracy, there is a clear benefit in utilizing the fine-scale details in the multiscale velocity fields when solving the transport equation.

5.3. Robustness with Respect to the Coarse Mesh

We now select a single layer (Layer 85) from the SPE10 model, and study how the multiscale methods perform for all possible coarse meshes with resolutions above a prescribed minimum. Figure 6 shows bar plots of the velocity errors for coarse-mesh resolutions larger than or equal to 5×11. The scale is chosen differently for each of the four plots
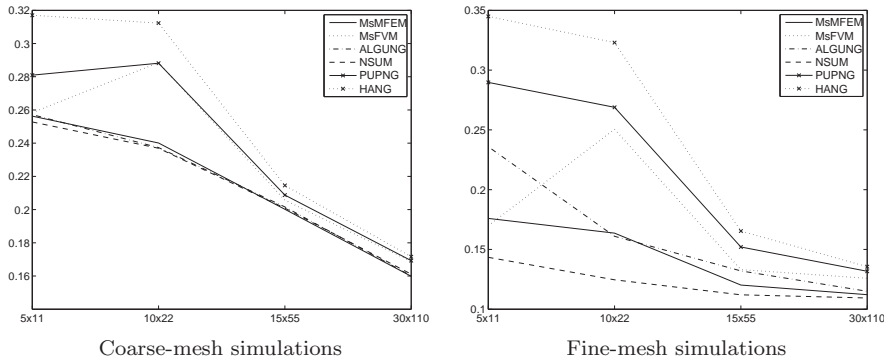
Figure 4. Multiscale and upscaling method saturation errors, $\delta(S)$, averaged over the smooth layers of the SPE10 model, for different coarse meshes, when solving the saturation equation on the coarse and fine meshes, respectively.
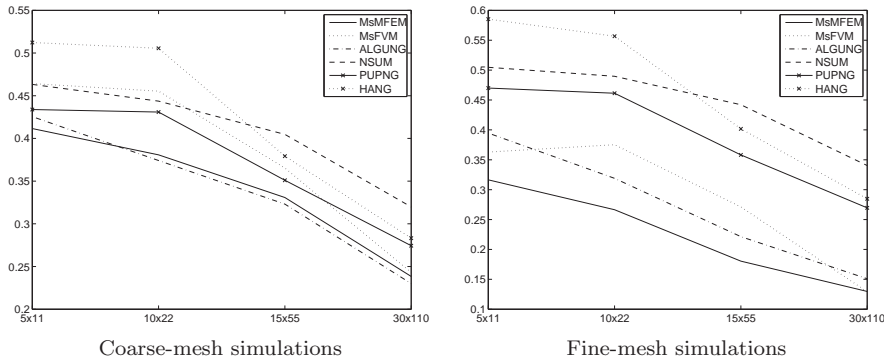

Figure 5. Multiscale and upscaling method saturation errors, $\delta(S)$, averaged over the channelized layers of the SPE10 model, for different coarse meshes, when solving the saturation equation on the coarse and fine meshes, respectively.

to emphasize the fact that all four methods loose accuracy when the aspect ratio of coarse-mesh elements becomes large. For MsFVM, the loss of accuracy is severe. This is due to the same issues causing large errors for a few of the layers in Section 5.1, since large aspect ratios in the mesh also cause large variations in pressure gradient sensitivity. This will be discussed more thoroughly in Section 5.5.

5.4. A Synthetic Test Suite

To perform a more systematic study of robustness, we consider three different heterogeneity scenarios and use the sequential Gaussian simulation routine `sgsim` from GSLIB [17] to generate 100 realizations of each. For each scenario, we apply the multiscale methods to two different source/sink configurations and measure velocity and satura-
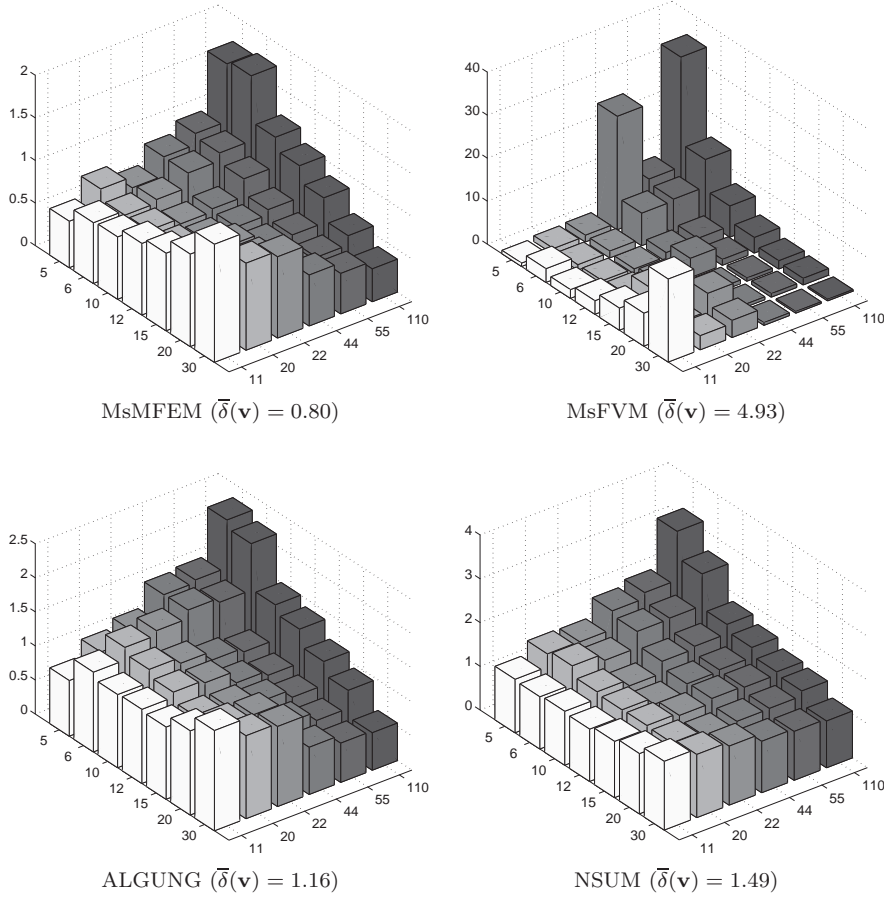
Figure 6. Velocity errors $\delta(\mathbf{v})$ as a function of coarse-mesh resolution for each of the multiscale methods. Also shown is the mean error $\overline{\delta}(\mathbf{v})$ over the displayed resolutions.

tions errors as we did above, except that we now average over the 100 different realisations. Each sample model consists of 64×64 square cells of unit size, and we consider four coarse meshes with unit aspect ratio: 4×4, 8×8, 16×16, and 32×32.

The first row of Figure 7 shows permeability fields for a typical realization of each scenario. Scenario 1 represents the common situation of log-normally distributed permeability with spatial correlation. Here the dimensionless correlation length equals 0.1 in both directions, and the variance of $\log \mathbf{K}$ is 5.0. The last two scenarios represent models with structures of very long correlation length, which are typically quite difficult to upscale. In Scenario 2, the structures run vertically and in Scenario 3 they are rotated $45°$ clockwise. The two different source/sink
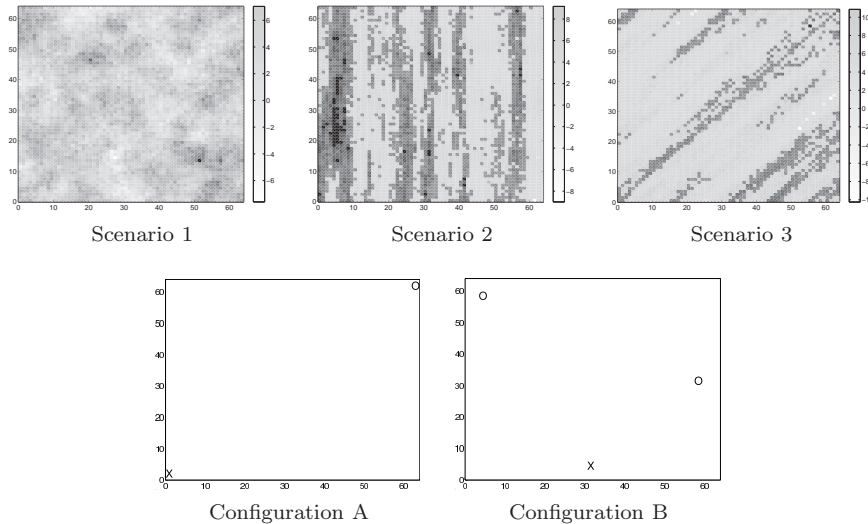
Figure 7. Permeability realizations from each of the three scenarios and the two different source/sink configurations; crosses represent sources and circles represent sinks.

configurations are shown in the second row of Figure 7. Configuration A is the standard quarter five-spot configuration.

Figure 8 shows the mean relative errors in velocity and saturation obtained by averaging over the 100 realizations for each scenario and source/sink configuration. The errors are shown as a function of the number of coarse-mesh cells in each direction. We note that velocity errors are larger than the saturation errors and generally lack the convergence behaviour of the latter. This is partly because the velocity typically oscillates rapidly as a response to the fine-scale heterogeneity.

The relative performance of the methods depends on both permeability data and source/sink configuration. The results reveal that all methods give reasonably accurate solutions for the spatially correlated log-normal permeability. For the scenarios with long correlation structures, we observe that MsFVM sometimes gives very inaccurate velocity fields, and that NSUM generally gives the least accurate saturation fields because of the limited ability to model variations in flow across coarse-mesh interfaces. In terms of saturation errors, MsMFEM is, on average, the most accurate method for all cases considered, except for Scenario 3A. In the following sections we will elaborate on special issues that cause the three multiscale methods to produce inaccurate results, and indicate possible remedies.

Although the ALGUNG method is not the most accurate method for any of the cases, it is generally quite robust in the sense that it
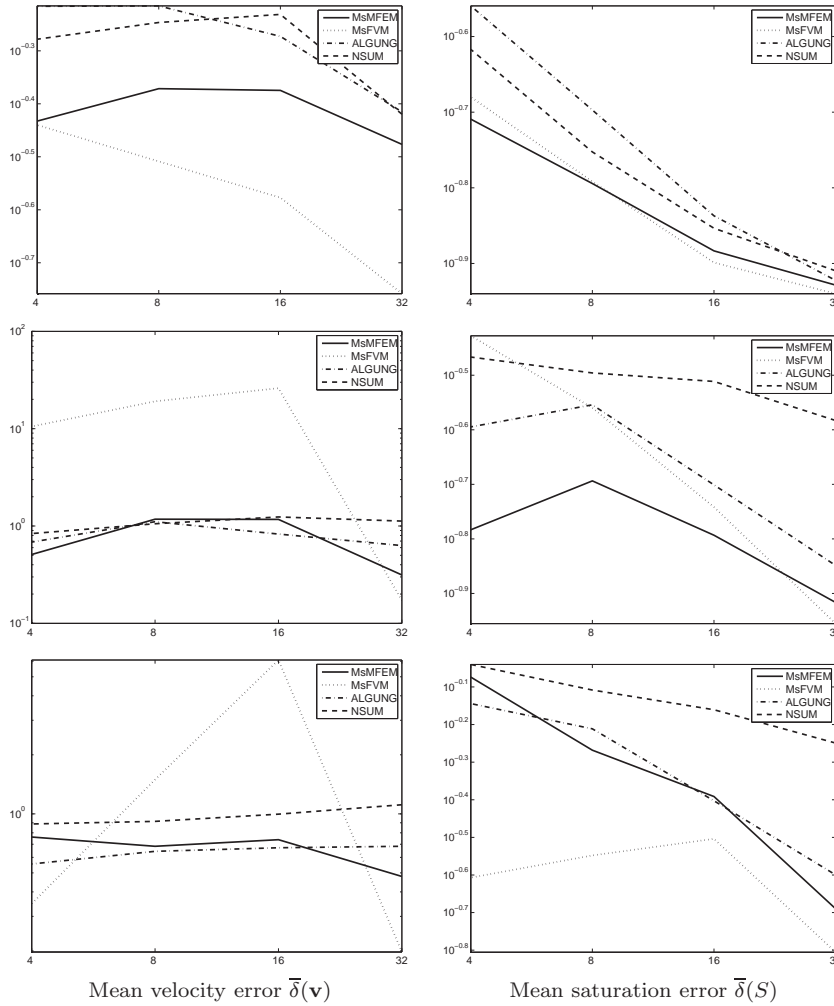
Figure 8. Mean errors as function of the number of coarse-mesh cells in each
direction for Configuration A for Scenarios 1 to 3 (from top to bottom).

never produces very poor solutions. The robustness of ALGUNG can
be attributed to the fact that the method invokes global information.
Indeed, ALGUNG employs the solution of a global flow problem to
determine boundary conditions for extended local upscaling. A direct
comparison of ALGUNG with the (local versions of the) three multi-
scale methods is therefore unjust in a certain sense, since ALGUNG
exploits more information about the flow than the three multiscale
methods. In Section 7.2 we discuss how global information can also be
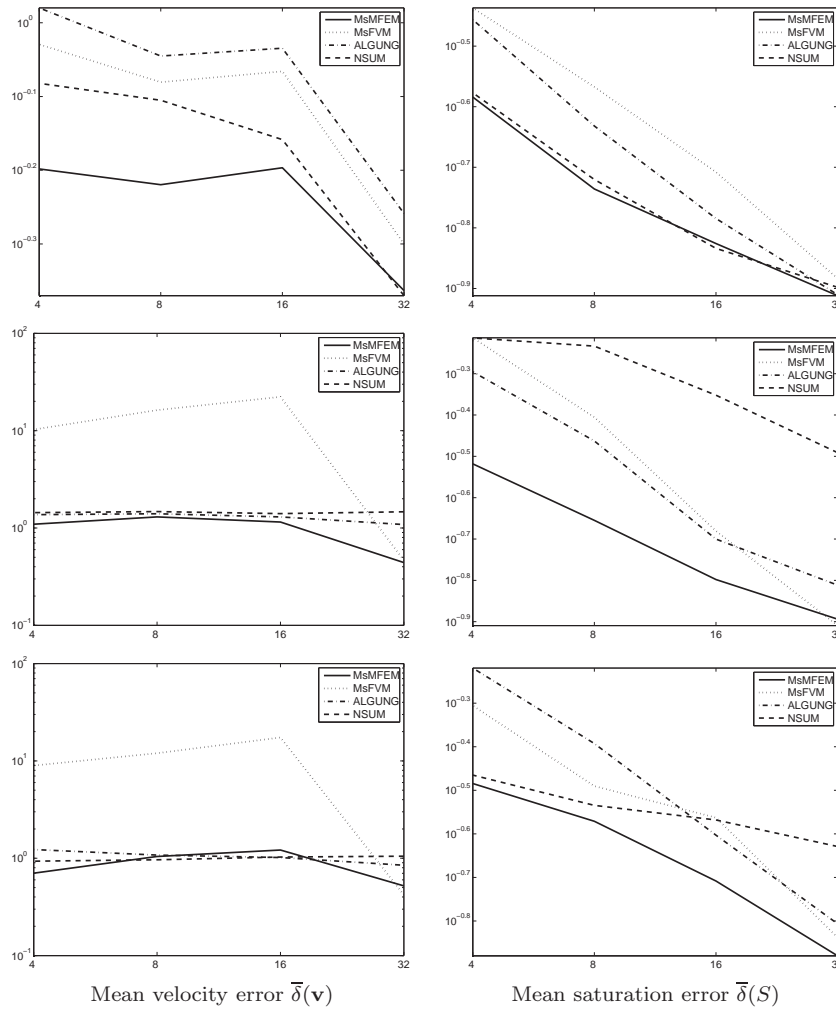incorporated into the three multiscale methods to further increase their

Figure 9. Mean errors as function of the number of coarse-mesh cells in each direction for Configuration B for Scenarios 1 to 3 (from to to bottom).

robustness for cases where local flow patterns are largely influenced by global effects.

5.5. MsFVM: High Aspect or Anisotropy Ratios

We have seen that MsFVM can produce inaccurate solutions, for instance when applied to meshes with high aspect ratios or for permeability fields with high-permeable channels on a low-permeable background. As an illustrative example, consider a realization from Scenario 1A above, i.e., a quarter five-spot with spatially correlated log-
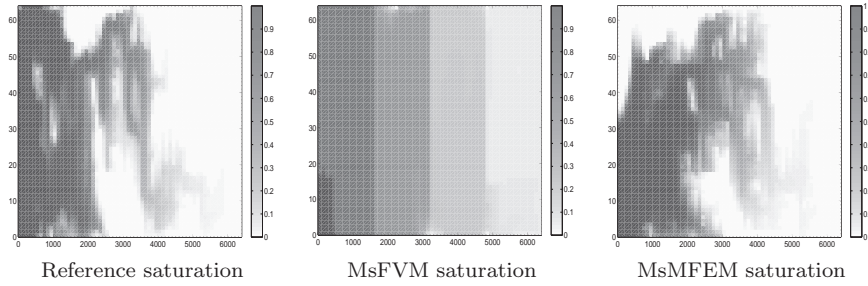
| Reference saturation | MsFVM saturation | MsMFEM saturation |

Figure 10. Saturation fields for a problem with high aspect ratios. For MsFVM the velocity error is $\delta(\mathbf{v}) = 353.00$ and the saturation error is $\delta(S) = 0.46$. For MsMFEM the velocity error is $\delta(\mathbf{v}) = 1.41$ and the saturation error is $\delta(S) = 0.57$.

normal permeability, but scale the model such that $\Delta x = 100\Delta y$. Figure 10 shows saturation profiles obtained from velocity fields computed using MsFVM and MsMFEM on a 4×4 coarse mesh together with a reference solution computed on the underlying 64×64 fine mesh. Although MsFVM produces a lower saturation error on the underlying fine mesh than MsMFEM for this problem, we note that all fine-scale details are lost in the MsFVM solution.

To understand why the saturation has been smeared out in Figure 10, consider the four basis functions supported in a single dual cell $\overline{E}$, e.g., the dark-shaded quadrant in Figure 1. The restriction of the MsFVM pressure solution to $\overline{E}$ can now be expressed as follows:

$$p|_{\overline{E}} = p(x_{i-1,j-1})\phi^{\overline{E}}_{i-1,j-1}$$
$$+ p(x_{i,j-1})\phi^{\overline{E}}_{i,j-1} + p(x_{i-1,j})\phi^{\overline{E}}_{i-1,j} + p(x_{i,j})\phi^{\overline{E}}_{i,j}.$$

Here $\phi^{\overline{E}}_{i,j}$ is the restriction of the basis function centred at $x_{i,j}$ to $\overline{E}$. Assuming that the permeability is smooth inside $\overline{E}$, we now deduce that each basis function is approximately bilinear and that the fluxes associated with each basis function are roughly ten thousand times larger in the y-direction than in the x-direction. Hence, if the pressure gradient in the y-direction is small, e.g., $p(x_{i-1,j-1}) \approx p(x_{i-1,j})$ and $p(x_{i,j-1}) \approx p(x_{i,j})$, then the total flux in the y-direction across the coarse-scale interfaces $\Gamma^{\overline{E}}$ inside $\overline{E}$ will be small relative to the flux contribution from each basis function. Thus, when we force mass conservation for a solution that is expressed as a linear superposition of the basis functions, we frequently get situations where the fine-scale fluxes are large relative to the associated coarse-scale flux, i.e, situations where

$$\int_{\Gamma^{\overline{E}}} |\mathbf{K}\nabla p \cdot \mathbf{n}| \; ds \gg \left| \int_{\Gamma^{\overline{E}}} \mathbf{K}\nabla p \cdot \mathbf{n} \; ds \right|.$$

$\phi_{i-1,j}^{\overline{E}}$-fluxes on $\Gamma_y^{\overline{E}}$ $\phi_{i-1,j-1}^{\overline{E}}$-fluxes on $\Gamma_y^{\overline{E}}$ $(\phi_{i-1,j-1}^{\overline{E}} - \phi_{i-1,j}^{\overline{E}})$-fluxes
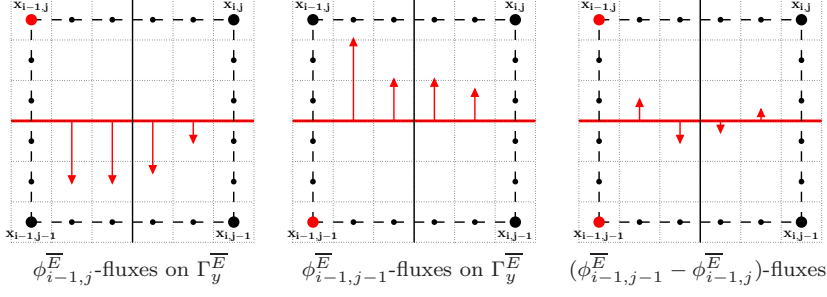
Figure 11. When adding the fluxes corresponding to $\phi_{i-1,j-1}^{\overline{E}}$ and $\phi_{i-1,j}^{\overline{E}}$, the sum of the fluxes along $\Gamma_y^{\overline{E}}$ is almost zero, but the subgrid fluxes may have roughly the same magnitude as the fluxes associated with each of the basis functions.
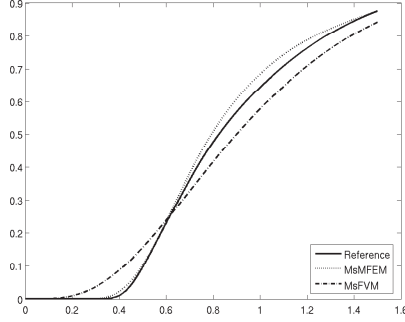

Figure 12. Water-cut curves measured for the problem with high aspect ratios.

This is because the contributions from $\phi_{i-1,j-1}^{\overline{E}}$ and $\phi_{i,j-1}^{\overline{E}}$ to the total flux across $\Gamma^{\overline{E}}$ approximately cancel the contributions from $\phi_{i-1,j}^{\overline{E}}$ and $\phi_{i,j}^{\overline{E}}$, respectively; see Figure 11. This implies that the boundary conditions used in the reconstruction step (19) are oscillatory, and hence that fine-scale fluxes across coarse-scale interfaces may oscillate.

Hence, since the fine-scale fluxes are used as boundary conditions for the subgrid problems in the final reconstruction step, the velocity fields that one obtains for this type of problem will typically contain circular currents that cause the saturation profiles to be smeared out. However, although the fine-scale velocity error is very large, the measured saturation error is often comparable to that of the other multiscale methods. This indicates that the problems that occur for MsFVM are fine-scale phenomena. Indeed, if we view MsFVM as an upscaling method and utilize only the coarse-scale fluxes to solve the saturation equation, then one generally obtains reasonable solutions.

Somewhat counter-intuitively, the smeared saturation profile in Figure 10 obtained using MsFVM is more accurate in the $\delta(S)$ measure
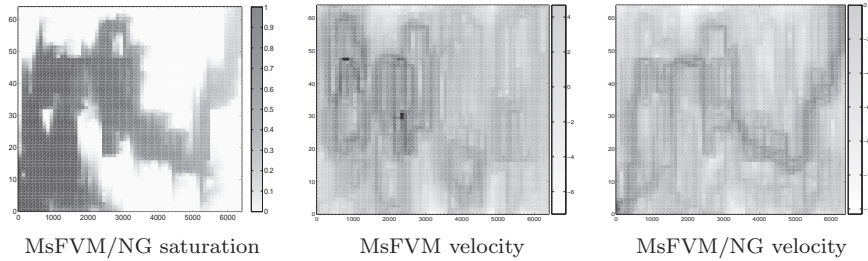
MsFVM/NG saturation   MsFVM velocity   MsFVM/NG velocity

Figure 13. Saturation and velocity fields for a problem with high aspect ratios obtained with MsFVM using the nested-gridding reconstruction. For comparison, we show also the MsFVM velocity field that contains circular currents. Using the nested-gridding reconstruction, the velocity error is reduced to $\delta(\mathbf{v}) = 1.97$, but the saturation error has increased to $\delta(S) = 0.63$.

than the corresponding solution obtained using MsMFEM. This indicates that $\delta(S)$ is not always an appropriate measure. Indeed, for real-life reservoir simulation one is more interested in computing functions of pointwise (or localized) saturations in wells, such as water cuts, cumulative production, etc. The MsMFEM saturation solution, which clearly seems to capture the qualitative behaviour of the saturation field better than MsFVM, gives significantly more accurate production characteristics; see Figure 12. Hence, in this case, errors in production characteristics better reflect the errors one expects from looking at the saturation profiles in Figure 10.

To improve the prediction of localized production characteristics, it is of interest to consider ways of removing the smearing effect that is sometimes experienced when using MsFVM. If the coarse-scale fluxes obtained using MsFVM are accurate, then the fine-scale solution will be accurate provided that the boundary conditions used in the final reconstructing step are appropriate. A natural approach to remove the smearing effect is therefore to replace the MsFVM reconstruction procedure with the nested-gridding reconstruction (24). Figure 13 shows that the circular currents in the velocity are removed using this option, but the saturation error has now increased slightly. Other possible approaches are to apply nested-gridding in certain regions only, or try to dampen the oscillations in the boundary conditions for the MsFVM reconstruction procedure. The latter approaches will typically introduce a problem-dependent parameter.

5.6. MsMFEM: Quarter Five-Spot with Diagonal Channels

In Section 5.4, we saw that Scenario 3A was the most difficult problem for MsMFEM. Since the method performed well on all cases for Scenario 3B (and on a large number of similar cases not reported here), it
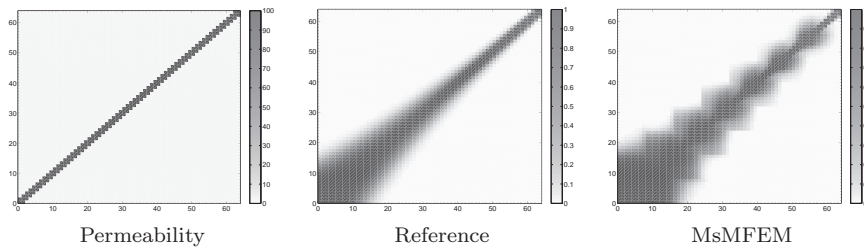
Figure 14. Reference and MsMFEM saturation profiles for a quarter-of-a-five-spot simulation of an idealized channel.
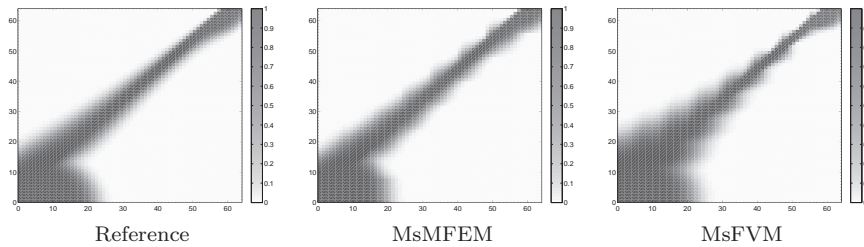

Figure 15. Reference, MsMFEM, and MsFVM saturation profiles for a quarter-of-a–five-spot simulation of the idealized channel in Figure 14, where the channel now has been moved slightly in the vertical direction.

is reasonable to claim that MsMFEM does not suffer from problems on flow cases where the major flow direction is not aligned with one of the coordinate directions. Rather, the difficulties of Scenario 3A are related to a special interplay between the quarter five-spot configuration, the diagonal structures, and the MsMFEM coarse mesh.

Figure 14 shows an idealized model with a narrow high-permeability channel along the diagonal. The permeability in the channel is 100 times the background permeability. We see that the MsMFEM saturation profile clearly bears the markings of the underlying 8×8 coarse mesh. To explain why the problem occurs, recall that the MsMFEM basis functions are designed to model inter-element flow, i.e., flow from one element to one of its neighbours. Since elements that share a common vertex are not considered to be neighbours, the flow we get by MsM-FEM is forced to take a detour into a neighbouring coarse element before continuing along the channel.

The situation in Figure 14 is a sort of worst-case scenario for MsM-FEM. Indeed, we have experienced mesh-related problems of this type only for models where a majority of the flow occurs in narrow channels that intersect corners of coarse elements. If we for instance perturb the synthetic model depicted in Figure 14 slightly so that the channel hits the midpoint of the element interfaces, then MsMFEM gives accurate

results, as shown in Figure 15. In the figure, we have also included the
corresponding MsFVM solution to illustrate that these mesh-related
problems are not particular to MsMFEM. The midpoints of the coarse-
mesh interfaces correspond to the corners of the dual coarse mesh used
for computing MsFVM basis functions, and the channel hitting these
corners results in MsFVM showing the same symptoms as MsMFEM
did on the original model.

5.7. NSUM: Enabling Variation Across Element Interfaces

To be able to compute subgrid contributions independently, NSUM had
to localize the fine-scale spaces to each coarse element. This localiza-
tion is a severe limitation when applying the method to datasets with
long correlation structures. Indeed, long and thin correlated structures,
such as the flow channels in the SPE10 model, cannot be accurately
represented with the BDM1 velocity basis functions on the coarse mesh.
MsMFEM, on the other hand, represents inter-element flow in a better
way, but is based on a low-order method and will therefore have lower
accuracy in smooth regions, as shown in [3].

Although NSUM and MsMFEM are based on fundamentally differ-
ent ideas, it is remarkably straightforward to combine the two methods
to obtain a scheme that is more general than MsMFEM, in the sense
that it allows higher-order methods on the coarse scale, and at the
same time captures inter-element flow better than NSUM. To this end,
we only need to replace the RT0 part of the NSUM velocity space
$\mathbf{V}_H$ by the MsMFEM velocity space $\mathbf{V}^{\mathrm{ms}}$. More precisely, we replace
the NSUM velocity approximation space $\mathbf{V}_{H,h}$ in (8) with the space
$\widetilde{\mathbf{V}}_{H,h} = \mathbf{V}_{H,h} - \mathbf{V}^{\mathrm{RT0}} + \mathbf{V}^{\mathrm{ms}}$.

We now make some observations that simplify the computations.
First, since the MsMFEM basis functions solve (2) locally, the fine-
scale responses $\mathbf{u}_h^j$ to these basis functions will be zero, as will the
response $\mathbf{u}_h^0$ to the fine-scale variation in source terms ($\mathcal{P}_{W_H} q$ in (13)
should be replaced by q). Thus, in addition to computing the MsMFEM
basis functions, we only need to solve local problems to compute the
responses associated with basis functions that are in $\mathbf{V}^{\mathrm{BDM1}}$, but not
in $\mathbf{V}^{\mathrm{ms}}$. The computational complexity of the combined approach will
therefore correspond roughly to the complexity of NSUM.

As we will see in Section 6 below, NSUM, and thus also the com-
bined NSUM–MsMFEM method, is significantly more expensive than
MsMFEM, but most of the computational time is spent solving local
problems. If we are willing to sacrifice some accuracy, we may obtain a
more efficient method by ignoring the nonzero fine-scale responses $\mathbf{u}_h^j$.
The overall method will then correspond to MsMFEM plus a BDM1

Table I. Velocity error $\delta(\mathbf{v})$ on the fluvial model for the two original and the two combined approaches.

	MsMFEM	NSUM	Ms-NSUM	Ms-BDM1
5×11	0.54533	1.22102	0.48941	0.52428
10×22	0.60181	1.24661	0.50085	0.57322
15×55	0.55667	1.38514	0.42369	0.47356
30×110	0.38635	1.06916	0.20954	0.23694

method on the coarse scale, and the computational complexity will be almost the same as for MsMFEM. In the following we refer to this latter approach as Ms-BDM1, and write Ms-NSUM when we speak of the approach where all fine-scale contributions are included.

To illustrate the performance of the combined approaches, we revisit the fluvial reservoir model from Section 5.1 (i.e., Layers 36–85 of the SPE10 model). The velocity errors $\delta(\mathbf{v})$ in Table I show the desired improvement of the NSUM method. For this problem, the effect of including all subgrid contributions, i.e., using Ms-NSUM instead of Ms-BDM1, does not appear to be significant. Moreover, since $\mathbf{V}^{\mathrm{ms}} \subset \widetilde{\mathbf{V}}_{H,h}$, the two combined methods give more accurate velocity fields than MsMFEM and the significance of the improvements increases with the resolution of the coarse mesh. For fields with smoother heterogeneity, numerous computer experiments not reported herein indicate that there may be more to gain by using a higher-order method on the coarse scale; see e.g., [26].

6. Computational Aspects

In this section we will discuss two important issues with the multiscale methods: computational efficiency and implementational aspects. For a discussion (mostly for MsMFEM) of two other important aspects—flexibility with respect to mesh types and cell geometries and flexibility with respect to subgrid solvers—we refer the reader to [3].

6.1. Computational Complexity

To discuss and compare the computational efficiency of the methods, we perform a simplified order-of-magnitude analysis under the assumption that the dominating factor is the solution of linear systems, with the time to solve a linear system of size $n \times n$ given by,

$$t(n) = \mathcal{O}(n^{\alpha}), \tag{27}$$

for some $\alpha > 1$. In particular, we ignore the work associated with determination of boundary conditions for local problems, numerical quadrature, and assembly of the linear systems.

The validity of these assumption may vary for the different methods and also depend on the size of the fine and coarse meshes. For large n, it is clear that (27) will dominate, but if the coarse mesh has nearly the same resolution as the fine mesh, the multiscale methods will need to solve many small linear problems. In this case, other factors may become significant, such as the work associated with numerical quadrature and assembly in NSUM, or the work associated with computing boundary conditions for MsFVM. However, only considering the solution of linear systems will give an idea of the potential efficiency of the methods.

For simplicity, we restrict our analysis to Cartesian meshes in D dimensions ($D = 2, 3$), and disregard special considerations at the global boundary $\partial\Omega$. Let N denote the total number of fine cells and divide the mesh into N_c coarse cells, each consisting of $N_s = N/N_c$ fine cells. Figure 16 shows a summary of the steps for each of the methods. For ALGUNG, m denotes the number of iterations needed to converge, r is the radius (with unit length equal to the coarse mesh parameter H) of the border region in the extended domains, and σ is the average fraction of coarse elements that need to be updated in each iteration. Estimates of the number of operations for each of the four methods are summarized in the tabular at the bottom of Figure 16.

To better understand the meaning of these estimates, we consider a specific model with a $128 \times 128 \times 128$ fine mesh. Figure 17 shows bar plots of the computational complexity as a function of coarse-mesh size, for two different values of α. Here it is assumed that ALGUNG converges in $m = 3$ iterations using border regions of size $r = 1/2$ (meaning that the boundary of the border region is obtained by connecting the cell centres of the nearest ring of surrounding coarse blocks, see Figure 2), and that on average $\sigma = 1/3$ of the transmissibility values needed to be updated in each iteration. We note that these assumptions generally yield a less accurate version of ALGUNG than the one we used above, since we then intended to show the potential accuracy of the method, whereas we now intend to show the potential efficiency. The plots indicate that MsMFEM and MsFVM are the most efficient methods, while NSUM and ALGUNG are more costly. Since NSUM has a significantly larger coarse-scale system than the other methods, its efficiency on high-resolution coarse meshes can be quite sensitive to the choice of linear solver. ALGUNG is more robust in this respect, but with the implementation that we have used to obtain the results in Section 5 it is by far the most expensive of the methods. However, as

MsMFEM:

1. Compute basis functions: $t_1 \approx D \cdot N_c \cdot t(2 \cdot N_s)$

2. Solve coarse-scale system: $t_2 \approx t(D \cdot N_c)$

NSUM:

1. Compute subgrid contribution: $t_1 \approx (2 \cdot D^2 + 1) \cdot N_c \cdot t(N_s)$

2. Solve coarse-scale system: $t_2 \approx t(D^2 \cdot N_c)$

MsFVM:

1. Calculate transmissibilities: $t_1 \approx 2^D \cdot N_c \cdot t(N_s)$

2. Solve coarse-scale system: $t_2 = t(N_c)$

3. Reconstruct conservative velocity field: $t_3 = N_c \cdot t(N_s)$

ALGUNG:

1. Initial local upscaling: $t_1 \approx D \cdot N_c \cdot t(N_s)$

2. For m iterations:

    a) Compute coarse-scale solution: $t_{2,a} = t(N_c)$

    b) Compute upscaled transmissibilities: $t_{2,b} \approx \sigma \cdot N_c \cdot t((2r + 1)^D \cdot N_s)$

3. Reconstruct fine-scale velocity field: $t_3 = N_c \cdot t(N_s)$

Method	Local (fine scale)		Global (coarse scale)
MsMFEM	$D \cdot 2^\alpha \cdot N_c \cdot N_s^\alpha$	$+$	$D^\alpha \cdot N_c^\alpha$
NSUM	$(2 \cdot D^2 + 1) \cdot N_c \cdot N_s^\alpha$	$+$	$D^{2 \cdot \alpha} \cdot N_c^\alpha$
MsFVM	$(2^D + 1) \cdot N_c \cdot N_s^\alpha$	$+$	N_c^α
ALGUNG	$(D + 1 + (2r + 1)^{\alpha \cdot D} \cdot \sigma \cdot m) \cdot N_c \cdot N_s^\alpha$	$+$	$m \cdot N_c^\alpha$

Figure 16. Summary of the simplified complexity analysis for the four methods.

mentioned in Section 4.2, a more efficient version of the algorithm can be obtained by choosing quite small border regions, defined in terms of rings of fine cells instead of rings of coarse cells [33]. Figure 17 also displays the cost of solving (2) on the fine scale using the TPFA method (16). It is clear that the efficiency of the multiscale methods relative to a direct fine-scale solution depends strongly on the computational complexity of the linear solvers available. For typical flow problems arising in porous media with highly oscillating coefficients, the algebraic multigrid method (AMG) [32] has reported complexity scaling as $\alpha \approx 1.2$, indicating that the situation in Figure 17b is realistic.
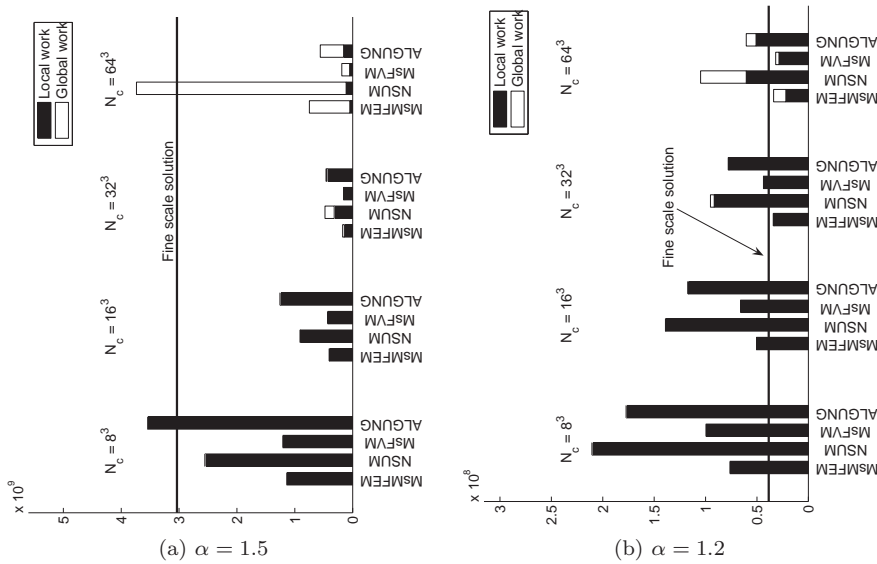
Figure 17. Comparison of computational work for the different methods as a function of coarse mesh size and the parameter α. For ALGUNG we have assumed that $\sigma = \frac{1}{3}$, $m = 3$ and $r = \frac{1}{2}$.

Figure 17b shows that, unless scale separation can be assumed, the computational complexity of the multiscale methods is comparable to the complexity of solving (2) directly on the fine scale using AMG. Multiscale methods are, however, not targeted at solving a single elliptic equation. Instead, the multiscale methods discussed herein are designed for dynamic problems where one needs to solve the same type of (elliptic) equation multiple times. This situation arises in, for instance, in two-phase and multiphase flow simulation. Then, the key to computational savings is that subgrid computations only need to be performed once, or infrequently throughout the simulations. Indeed, Figure 17 shows that almost all of the computational work in the multiscale methods is associated with solving local fine-scale problems. It should also be noted that multiscale methods, as opposed to algebraic multigrid algorithms, are almost trivial to parallelize, and can also be used to decompose problems that are too large to fit into memory.

6.2. COMMENTS ON IMPLEMENTATION

During the last few years we have made several implementations (and careful reimplementations) of various multiscale and upscaling-downscaling methods, amongst others the four methods discussed above. Through this work, we have gained valuable insight into several issues that affect

how difficult the various methods are to implement. A discussion of the implementational complexity of one method relative to another will (of course) never be completely unbiased. Nevertheless, we will here try to point out some issues, since the complexity of making an (efficient and flexible) implementation is often difficult to read from the scientific papers describing the method.

All methods are quite straightforward to implement for cases where the coarse and the fine mesh are Cartesian. However, in our opinion the finite-element methods are less difficult to implement. The domains for computing pressure basis functions for MsFVM and the domains used for the extended local upscaling in ALGUNG are generally chosen differently along external boundaries and in the interior of the global domain. This introduces special cases along external boundaries for both MsFVM and ALGUNG. In addition, we claim that the need to introduce a dual mesh complicates the implementation of MsFVM. These issues are avoided in the multiscale methods based on a finite-element formulation.

Additional complications arise for complex meshes, for instance, for unstructured meshes and non-conforming meshes with non-matching faces (that typically appear in geological models along large-scale fractures and faults). For NSUM it may be difficult to define an approximation space $\mathbf{V}_H$ on the coarse mesh; using an RT0 space or a BDM1 space will generally place strong restrictions on the coarse mesh. For MsFVM it may be difficult to define a dual mesh and to generate proper boundary conditions for the pressure basis functions. For ALGUNG it may be difficult to define proper interpolation operators that map coarse-scale pressure solutions onto the boundary of regions for extended local upscaling. These complicating factors are avoided in MsMFEM. Hence, based on our experience we strongly believe that MsMFEM is less complicated to implement, especially for complex meshes. Indeed, the cells in the coarse mesh can in principle consist of an arbitrary connected collection of cells from the fine mesh. In fact, given a proper fine-scale solver on the fine mesh, the implementation of multiscale methods based on finite elements is relatively straightforward, as shown in [3], regardless of whether the underlying fine-grid solver uses a finite-element or a finite-volume formulation.

Another important question is the ease with which one can include more complex flow physics. Here, however, it is not possible to make an assessment of the various methods, since MsFVM, so far, is the only method that has been extended to more complex physical models such as the black-oil model.

7. Two-Phase Flow Simulations

In Section 5 we demonstrated that given an elliptic problem of the form (2), the multiscale methods introduced in Sections 3 and 4 may be used as approximate fine-scale solution methods. Moreover, in the previous section we discussed the computational complexity for a *single* pressure solution and concluded that the multiscale methods are not significantly faster than solving the fine-scale problem directly with a (very) efficient linear solver. We also remarked that multiscale methods for flow in porous media are targeted at multiphase flow applications where one needs to solve an equation of the form (2) multiple times.

In this section we use the multiscale methods to solve time-dependent pressure equations as part of incompressible two-phase simulations, for which the pressure will depend on the saturation distribution through λ_t and therefore needs to be recomputed repeatedly throughout the simulation. The simulations will be performed on Layer 85 from the SPE10 model; this is a fluvial layer where MsFVM does not suffer from the problems described in Section 5.5. Source/sink terms are placed in quarter five-spot pattern, and we define mobilities by

$$\lambda_w = MS^2, \qquad \lambda_o = (1-S)^2, \qquad 0 \le S \le 1.$$

Hence, we use quadratic relative permeabilities and assume unit oil viscosity. The coarse mesh used in all simulations below is 10×22.

To solve the system (2)–(3), we apply a non-iterated sequential splitting. This means that the pressure equation is solved at the current time-step with total mobility computed using saturations from the previous time-step. Next, the the saturations are convected forward in time using the current velocities, and the new saturation values are used to compute the pressure at the next time-step, and so on.

The dynamic character of an incompressible two-phase flow system is often quantified by the ratio of the end-point values of the total mobility, usually called the mobility ratio for brevity. In our case, we have $\lambda_t(0) = 1$ and $\lambda_t(1) = M$, so that the end-point mobility ratio for the problems considered in this section is M. Different end-point mobility ratios give rise to very different flow scenarios. Indeed, with $M > 1$ we get so-called unstable displacement flows, for which small-scale fingers develop and move rapidly at the saturation front. In contrast, $M < 1$ gives so-called stable displacement flows, for which the saturation front is quite sharp (piston-like displacement). We will apply the multiscale methods to both stable and unstable displacement flows to demonstrate their versatility.

The key to get enhanced efficiency when using multiscale methods is to perform new subgrid computations only where it is necessary, e.g., in

regions where large changes in the total mobility λ_t have occurred. The main objective in this section is to reveal to what extent the subgrid computations in the respective multiscale methods can be avoided. To this end, we only consider MsMFEM, MsFVM, and Ms-NSUM (the version of NSUM that is modified to enable variation across coarse-mesh interfaces). We apply Ms-NSUM instead of NSUM because in this framework the key to obtain accurate simulation results with low computational cost is to build as much information as possible into the initial approximation space $\mathbf{V}_{H,h}$.

There are two reasons why we do not consider ALGUNG. First, to avoid the full iterative algorithm at each time-step, ALGUNG needs to be supplied with a strategy for updating transmissibilities when the total mobility changes. This can be done by, e.g., using averaged values of the total mobility, but such approaches are, to our knowledge, not well-documented in the literature. Secondly, to obtain a mass-conservative velocity field on the fine mesh, one needs to perform the nested-gridding reconstruction at each time-step. Hence, even if we have a strategy for updating the coarse-scale transmissibilities at each time step, ALGUNG still needs to perform subgrid computations everywhere.

7.1. Efficient Two-Phase Flow Implementations

The purpose of this section is to demonstrate that one may obtain accurate two-phase flow simulation results using MsMFEM, MsFVM, and Ms-NSUM without recomputing the multiscale basis functions, i.e., the velocity basis functions $\mathbf{\Psi}_{ij}$ that span $\mathbf{V}^{\mathrm{ms}}$ for MsMFEM and Ms-NSUM, and the pressure basis functions ϕ_j for MsFVM. For MsM-FEM and Ms-NSUM one only needs to reassemble the mass matrix that stems from the term $(\mathbf{K}^{-1}\mathbf{u}_h, \mathbf{v}_h)$ in (5) (but with $\mathbf{K}$ replaced with $\mathbf{K}\lambda_t$), and solve the corresponding coarse-scale system. Hence, all subgrid computations that involve solving local elliptic problems are made part of a preprocessing step. The reassembly of the mass matrix involves only local matrix-vector products and is therefore a computation with $\mathcal{O}(n)$ complexity, where n is the number of cells in the fine mesh.

For MsFVM it is not possible to avoid all subgrid computations without also suffering from severe loss of accuracy. Indeed, if all subgrid computations were to be made part of a preprocessing step, one could neither recompute the basis functions nor repeat the final reconstruction step (19). To achieve this, one could not update the coarse-scale transmissibilities. This implies that the coarse-scale solution would not change (unless the source and sink terms change) throughout the sim-

ulation, i.e., one would obtain a static velocity field. This is a severe limitation that generally gives a major loss of accuracy. Hence, in the simulations reported below where we state that we do not update basis functions, we do not recompute the pressure basis functions, but we *do* update the coarse-scale transmissibilities according to changes in the total mobility. We therefore need to perform the reconstruction step (19) in all cells in the coarse mesh at every time-step.

As discussed in Section 5.5, the L^2-saturation error (26) is not always an applicable measure for assessing the predictive qualities of a numerical two-phase flow model. In addition to the saturation errors, we therefore consider errors in the water-cut $w(t)$; i.e., errors in the fraction of water in the produced fluid as a function of time. We measure errors in the water-cut using the following measure:

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

Figure 18 shows water-cut curves and corresponding saturation errors for a stable displacement with $M = 0.1$. We observe that the accuracy is slightly improved by recomputing the basis functions at each time step. Moreover, we see that the methods based on finite elements give larger errors in water-cut than MsFVM when the basis functions are never updated. This is to be expected since MsFVM reconstructs the conservative fine-scale velocity field in every step, and is therefore able to account for the large variations in total mobility as a strong saturation front passes through a coarse cell.

Next we consider mobility ratio $M = 10$, giving an unstable displacement. Figure 19 demonstrates that unstable displacement flow scenarios are actually simpler for the multiscale methods. Indeed, all methods give highly accurate saturation solutions without recomputing basis functions. For high end-point mobility ratios, the displacement profile consists of a weak water shock followed by a smooth and slowly increasing rarefaction wave. In other words, there are no abrupt changes in total mobility within coarse cells, and it is therefore sufficient to account for mobility changes on the coarse scale. This is an advantage for the methods based on finite elements, since they do not need to perform subgrid calculations at all in these situations.

Adaptive Simulation with MsFVM

In the previous section we remarked that it is not possible to make all subgrid computations for MsFVM part of a preprocessing step without significant loss of accuracy. Clearly, this makes MsFVM much less efficient than the corresponding versions of MsMFEM and Ms-NSUM. It has been shown [24, 23] that MsFVM can be made more efficient by selectively updating basis functions and coarse-scale transmissibilities
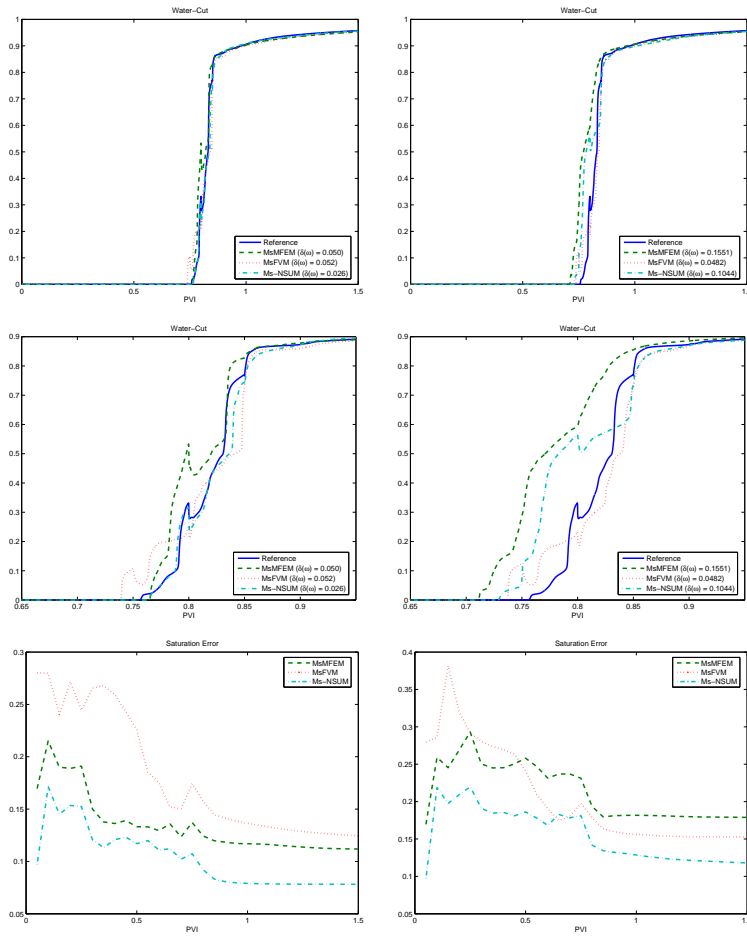
Figure 18. Saturation errors and water-cuts for the whole production period and zoomed in around the water breakthrough for mobility ratio $M = 0.1$. The basis functions are recomputed at each time step (left) and never recomputed (right).
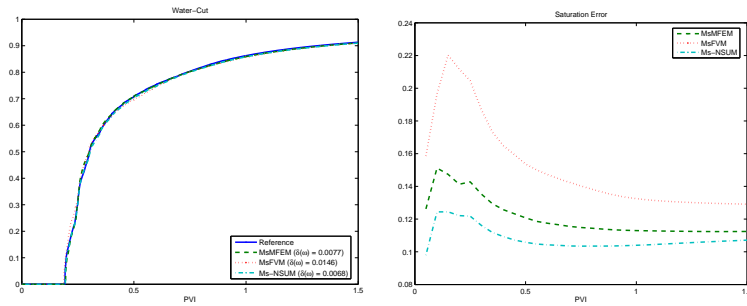

Figure 19. Water-cuts and saturation errors for mobility ratio $M = 10$. The basis functions are only computed initially.

only in regions with large changes in mobility. In this approach, one introduces a second set of basis functions to express the reconstructed velocity field as a linear superposition of basis functions, i.e., analogously to the way the fine-scale velocity is obtained in MsMFEM and Ms-NSUM. Unfortunately, as explained in Section 4.1, the number of additional basis functions needed is quite large. A direct reconstruction at every step may therefore be more efficient unless the fraction of updated basis functions is very small [23].

The authors have observed that it tends to be quite difficult to make the adaptive version of MsFVM robust; by recomputing only a portion of the basis functions and coarse-scale transmissibilities, one frequently obtains inaccurate results. Hence, to avoid severe loss of accuracy, we recompute all coarse-scale transmissibilities at each time step, and thereby perform a direct reconstruction using (19) in all cells in the coarse mesh at each time-step. To make this version of MsFVM as efficient as possible, we do not recompute the pressure basis functions, since these generally have less impact on the results.

Adaptive Simulation with MsMFEM and Ms-NSUM
For mobility ratio $M < 1$, i.e., for cases where the viscosity of the wetting fluid is larger than the viscosity of the non-wetting fluid, it was observed in [1] that enhanced accuracy can be obtained by recomputing basis functions in regions where the total mobility has changed significantly since the previous update. In particular, it was proposed that all basis functions that are supported in a coarse-mesh element E_i with

$$\frac{1}{|E_i|} \int_{E_i} \lambda_t(x) \, \mathrm{d}x > \epsilon,$$

for some threshold ϵ should be recomputed. In Figure 20 we show the results obtained with this approach for $M = 0.1$ and $\epsilon = 0.2$, which corresponds to updating on average 10% of the basis functions.

Comparing with the results depicted in Figure 18 we see that one obtains significantly more accurate water-cut curves for both MsMFEM and Ms-NSUM using the adaptive strategy. However, it should be noted that the cost of reconstructing on average 10% of the basis functions is about half the computational cost of performing the reconstruction (19) for MsFVM in all blocks. The adaptive version of MsMFEM and Ms-NSUM is therefore only slightly less expensive than MsFVM with no pressure basis functions recomputed.
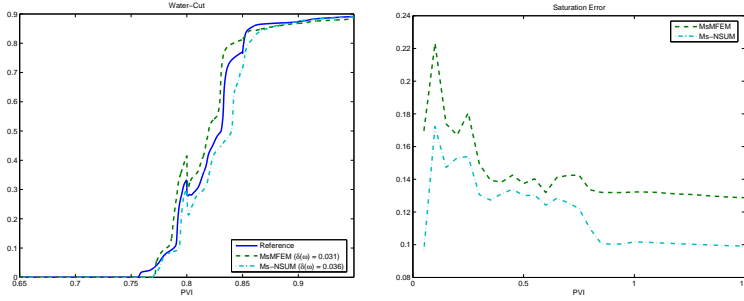
Figure 20. Water-cuts zoomed in around the water breakthrough and saturation errors for the finite-element based multiscale methods when updating basis functions near the saturation front. The mobility ratio is $M = 0.1$, and on average 10% of the basis functions are updated.

7.2. Utilizing an Initial Fine-Scale Velocity Solution

In Section 5.4 we argued that the relatively robust behaviour of the ALGUNG method with respect to different flow configurations could be attributed to the use of global information to build local boundary conditions. From this observation, we may find it natural to ask if the accuracy of the multiscale methods can be improved by building more information into the local subgrid problems. For instance, is it possible to incorporate global effects that stem from global boundary conditions, well configurations, and large correlation lengths as seen in reservoirs with channelized heterogeneity structures?

For MsMFEM and MsFVM it was demonstrated in [1] and [19], respectively, that it is indeed possible to exploit global information from an initial fine-scale solution to enhance accuracy. For MsMFEM, this is done by adding an extra "boundary" condition in (6). If $\mathbf{v}$ is the initial fine-scale velocity solution, one requires that

$$\psi_{ij} \cdot \mathbf{n}_{ij} = \frac{\mathbf{v} \cdot \mathbf{n}_{ij}}{\int_{\Gamma_{ij}} \mathbf{v} \cdot \mathbf{n}_{ij}\, ds} \qquad \text{on } \Gamma_{ij},$$

and thereby splits the calculation of basis functions into two separate problems in E_i and E_j, respectively. A corresponding version of Ms-NSUM is obtained by replacing the RT0 part of $\mathbf{V}_H$ with the span of the basis functions used in the improved MsMFEM.

Similarly, an initial global fine-scale pressure solution p may be used to specify boundary conditions for the pressure basis functions in MsFVM. To specify the basis function $\phi_{i,j}$ associated with node $x_{i,j}$
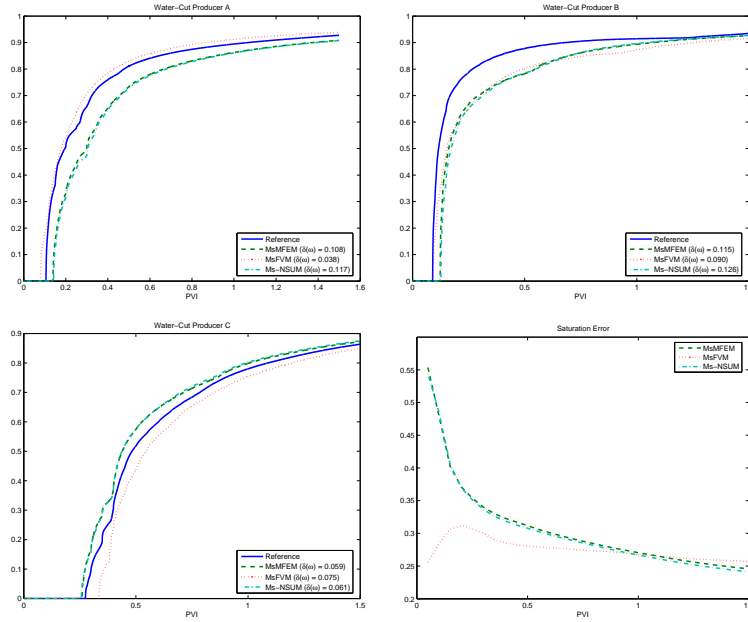
Figure 21. Water-cuts and saturation errors for MsMFEM, MsFVM and Ms-NSUM
on Layer 68 from the SPE10 model when using standard boundary conditions for
the basis functions.

inside the dark-shaded quadrant in Figure 1, one requires that

$$b(x) = \begin{cases} \frac{p(x)-p(x_{i,j-1})}{p(x_{i,j})-p(x_{i,j-1})} & \text{on } \Gamma_2, \\ \frac{p(x)-p(x_{i-1,j})}{p(x_{i,j})-p(x_{i-1,j})} & \text{on } \Gamma_4. \end{cases}$$

As before, homogeneous boundary conditions are used on Γ_1 and Γ_3.

To demonstrate the utility of these approaches, we consider Layer
68 from the SPE10 model, which is quite difficult for all of the multi-
scale methods according to Figure 3. In particular, this was the only
layer where MsMFEM produced saturation error significantly above
the level of its general performance. We use the experimental setup
from Section 5.1, i.e., five-spot source configuration and a 15×55
coarse mesh, and choose a mobility ratio of $M = 10$ to eliminate issues
related to updating of basis functions from the discussion. Figure 21
shows the saturation error and water-cuts for the three producers where
there were significant errors, when only local information is used in the
multiscale solutions. For MsMFEM and Ms-NSUM, the large saturation
errors initially indicate that some trends are not appropriately captured
by the finite-element basis functions. The corresponding results when
utilizing an initial fine-scale solution are displayed in Figure 22 and
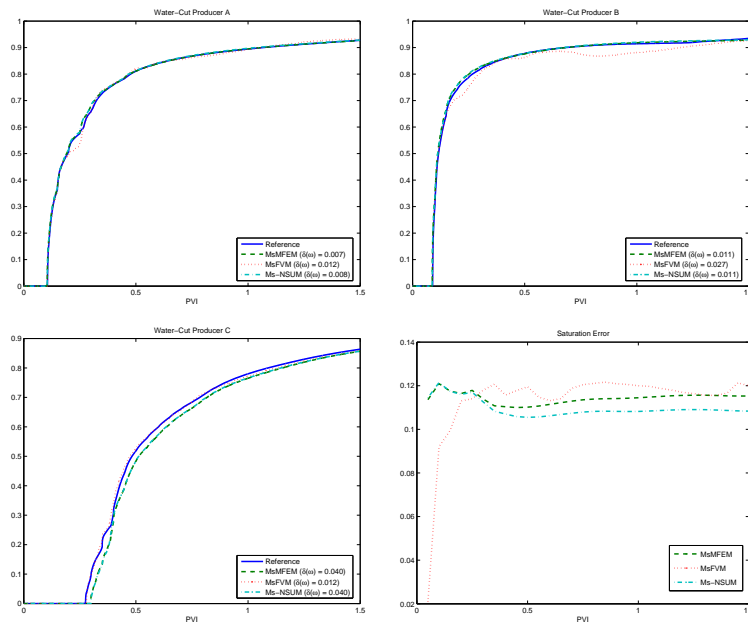demonstrate improved overall accuracy for all of the methods.

Figure 22. Water-cuts and saturation errors for MsMFEM, MsFVM and Ms-NSUM on Layer 68 from the SPE10 model when global information is used to define boundary conditions for the basis functions.

The results obtained here reflect that by incorporating global information from an initial fine-scale solution one obtains methods capable of capturing effects that normally are not resolved properly, such as large-scale flow barriers (see [2] for a discussion of MsMFEM's accuracy around low permeable obstacles). In fact, we find invariably that the saturation and water-cut errors are considerably reduced when initial fine-scale solutions are utilized. We would also like to point out that utilizing an initial fine-scale solution alleviates the problems in Sections 5.5 and 5.6 for MsFVM and MsMFEM, respectively. However, for MsFVM we have encountered examples where the version using an initial fine-scale solution introduces velocity instabilities even though the original approach does not, and we can therefore only recommend it as a complement to other stabilization procedures.

As discussed in Section 6, obtaining a single fine-scale solution is in practice less computationally demanding than constructing the full set of basis functions. Hence, provided the global flow patterns do not change significantly during the simulation, utilizing an initial fine-scale solution can increase the accuracy at a relatively low computational overhead. However, if flow conditions change frequently, e.g., due to shut down of wells, the local approach may be preferable. Alterna-

tively, one could exploit information from initial fine-scale solutions from generic, axes-oriented flows, similar in spirit to what was done in the original local-global upscaling method [15].

8. Concluding Remarks

In this paper we have reviewed three multiscale methods applicable to elliptic problems in porous media flow; the two-scale conservative subgrid method (NSUM) [5, 6]; the multiscale mixed finite-element method (MsMFEM) [13, 1]; and the multiscale finite-volume method (MsFVM) [25, 24]. As a benchmark for the multiscale methods, we use the adaptive local-global upscaling method [14] combined with a nested-gridding downscaling approach (ALGUNG), which is a state-of-the-art upscaling-downscaling approach with multiscale characteristics. The performance of the methods is compared in terms of accuracy, robustness, computational complexity, implementational complexity, and the ability to accelerate fine-scale two-phase flow simulations.

The results presented in the paper demonstrate that the multiscale methods are quite robust when applied to problems with spatially correlated log-normal permeability fields. For channelized permeability fields, so-called fluvial formations, the accuracy of the methods differ. MsMFEM and ALGUNG appear to be robust and produce accurate solutions. NSUM generally gives lower accuracy because it is not able to resolve the channels correctly. Finally, MsFVM can in some cases produce velocity solutions with local circular currents. For these cases, MsFVM produces solutions that are reasonable on a coarse-scale, but fails to give correct flow patterns on a fine scale. We presented ways of remedying the shortcomings of NSUM and MsFVM. We also showed that MsMFEM (and MsFVM) can suffer from loss of accuracy due to grid effects in a special case.

To assess the computational complexity of the methods, we have presented complexity estimates that, along with our own experience, indicate that MsMFEM and MsFVM are less expensive than NSUM and ALGUNG. The latter methods can, however, be made more efficient by sacrificing some accuracy, where the gain in computational efficiency is significantly larger than the accuracy loss.

In our experience, all four methods are more or less equally straightforward to implement for Cartesian meshes, although MsFVM and ALGUNG have some mesh-related issues that make the resulting codes more voluminous. For unstructured meshes, meshes with non-matching faces, or meshes with complex cell geometries, we claim that MsM-

FEM is significantly less complicated to implement (see e.g., [3] for implementations of MsMFEM on corner-point grids).

The multiscale methods considered in this paper can be viewed both as upscaling methods and as approximate fine-scale solvers. As upscaling methods they generally offer equally or more accurate solutions than traditional upscaling methods. As approximate fine-scale solvers, they offer a tool to compute high-resolution velocity fields for dynamic flow simulations at a low computational cost.

In particular it is shown that using MsFVM, MsMFEM and a special version of NSUM, it is possible to obtain accurate two-phase flow simulations without performing all the local subgrid computations at each time step. For MsMFEM it is even possible to get accurate results by performing the subgrid computations as part of a preprocessing step. For MsFVM similar results can be obtained by performing only a small portion of the subgrid computations during the simulation. In this way, these multiscale methods offer a tool to accelerate simulations of multiphase flow on high-resolution geological models, and can therefore bring the industry closer to having an Earth Model shared between the reservoir engineers and the reservoir geologists.

Acknowledgements

References

1. 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.
2. 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.
3. Aarnes, J. E., S. Krogstad, and K.-A. Lie: 2006, submitted, 'Multiscale mixed/mimetic methods on corner-point grids'. *Comput. Geosci.*
4. Aarnes, J. E. and K.-A. Lie: 2004, 'Toward reservoir simulation on geological grid models'. In: *Proceedings of the 9th European Conference on the Mathematics of Oil Recovery*. Cannes, France.
5. Arbogast, T.: 2000, 'Numerical subgrid upscaling of two-phase flow in porous media'. In: *Numerical treatment of multiphase flows in porous media (Beijing, 1999)*, Vol. 552 of *Lecture Notes in Phys.* Berlin: Springer, pp. 35–49.

6.   Arbogast, T.: 2002, 'Implementation of a locally conservative numerical subgrid
     upscaling scheme for two-phase Darcy flow'. *Comput. Geosci.* **6**(3-4), 453–481.

7.   Arbogast, T.: 2004, 'Analysis of a two-scale, locally conservative subgrid
     upscaling for elliptic problems'. *SIAM J. Numer. Anal.* **42**(2), 576–598.

8.   Arbogast, T., S. E. Minkoff, and P. T. Keenan: 1998, 'An operator-based
     approach to upscaling the pressure equation'. In: V. N. B. et al. (ed.): *Computational Methods in Water Resources XII*, Vol. 1. Southampton, U.K., pp.
     405–412.

9.   Brezzi, F.: 2000, 'Interacting with the subgrid world'. In: *Numerical analysis
     1999 (Dundee).* Chapman & Hall/CRC, Boca Raton, FL, pp. 69–82.

10. Brezzi, F. and M. Fortin: 1991, *Mixed and hybrid finite element methods*, Vol. 15
     of *Springer Series in Computational Mathematics.* New York: Springer-Verlag.

11. Brezzi, F., J. D. Jr., and L. D. Marini: 1985, 'Two families of mixed elements
     for second order elliptic problems'. *Numer. Math.* **47**, 217–235.

12. Cardwell, W. and R. L. Parsons: 1945, 'Avearge permeabilities of heterogeneous
     oil sands'. *Trans. Am. Inst. Mining. Met. Pet. Eng.* pp. 34–42.

13. Chen, A. and T. Hou: 2002, 'A mixed multiscale finite element method for
     elliptic problems with oscillating coefficients'. *Math. Comp.* **72**(242), 541–576.

14. Chen, Y. and L. J. Durlofsky: 2006, 'Adaptive local-global upscaling for general
     flow scenarios in heterogeneous formations'. *Transp. Porous Media* **62**(2), 157–
     182.

15. Chen, Y., L. J. Durlofsky, M. Gerritsen, and X. H. Wen: 2003, 'A coupled
     local-global upscaling approach for simulating flow in highly heterogeneous
     formations'. *Adv. Water Resour.* **26**(10), 1041–1060.

16. Christie, M. A. and M. J. 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.

17. Deutsch, C. V. and A. G. Journel: 1998, *GSLIB: Geostatistical software library
     and user's guide.* New York: Oxford University Press, 2nd edition.

18. Durlofsky, L.: 1991, 'Numerical calculations of equivalent gridblock permeability tensors for heterogeneous porous media'. *Water Resour. Res.* **27**(5),
     699–708.

19. Efendiev, Y., V. Ginting, T. Hou, and R. Ewing: 2006, 'Accurate multiscale
     finite element methods for two-phase flow simulations'. *J. Comput. Phys.*
     220(1), 155–174.

20. Gautier, Y., M. J. Blunt, and M. A. Christie: 1999, 'Nested gridding and
     streamline-based simulation for fast reservoir performance prediction'. *Comput.
     Geosci.* **3**(3–4), 295–320.

21. Hou, T. Y. and X.-H. Wu: 1997, 'A multiscale finite element method for elliptic
     problems in composite materials and porous media'. *J. Comput. Phys.* **134**(1),
     169–189.

22. Hughes, T., G. Feijoo, L. Mazzei, and J. Quincy: 1998, 'The variational multiscale method - a paradigm for computational mechanics'. *Comput. Methods
     Appl. Mech. Engrg* **166**, 3–24.

23. Jenny, P., S. Lee, and H. Tchelepi: 2006, 'Adaptive fully implicit multi-scale
     finite-volume methods for multi-phase flow and transport in heterogeneous
     porous media'. *J. Comput. Phys.*

24. Jenny, P., S. H. Lee, and H. A. Tchelep: 2004/05, 'Adaptive multiscale finite-
     volume method for multiphase flow and transport in porous media'. *Multiscale
     Model. Simul.* **3**(1), 50–64.

25. Jenny, P., S. H. Lee, and H. A. Tchelepi: 2003, 'Multi-scale finite-volume method for elliptic problems in subsurface flow simulation'. *J. Comput. Phys.* **187**(1), 47–67.

26. Kippe, V., J. Aarnes, and K.-A. Lie: 2006, 'Multiscale finite-element methods for elliptic problems in porous media flow'. In: *CMWR XVI — Computational Methods in Water Resources, Copenhagen, Denmark, June, 2006.* `http://proceedings.cmwr-xiv.org/`.

27. Lunati, I. and P. Jenny: 2006, 'Multiscale finite-volume method for compressible multiphase flow in porous media'. *J. Comput. Phys.* **216**(2), 616–636.

28. Matache, A.-M. and C. Schwab: 2000, 'Homogenization via p-FEM for problems with microstructure'. In: *Proceedings of the Fourth International Conference on Spectral and High Order Methods (ICOSAHOM 1998) (Herzliya)*, Vol. 33. pp. 43–59.

29. Raviart, P.-A. and J. M. Thomas: 1977, 'A mixed finite element method for 2nd order elliptic problems'. In: *Mathematical aspects of finite element methods (Proc. Conf., Consiglio Naz. delle Ricerche (C.N.R.), Rome, 1975)*. Berlin: Springer, pp. 292–315. Lecture Notes in Math., Vol. 606.

30. Renard, P. and G. de Marsily: 1997, 'Calculating equivalent permeability: a review'. *Adv. Water Resour.* **20**(5–6), 253–278.

31. Sangalli, G.: 2003, 'Capturing small scales in elliptic problems using a residual-free bubbles finite element method'. *Multiscale Model. Simul.* **1**(3), 485–503.

32. Stüben, K.: 2000, *Multigrid*, Chapt. Algebraic Multigrid (AMG): An Introduction with Applications. Academic Press.

33. Wen, X., L. Durlofsky, and Y. Chen: 2005, 'Efficient three-dimensional implementation of local-global upscaling for reservoir simulation'. In: *SPE Symposium of Reservoir Simulation, Houston, TX, January 31 Â February 2, 2005.* SPE 92965.