SPE 141473

Two-Stage Algebraic Multiscale Linear Solver for Highly Heterogeneous Reservoir Models

H. Zhou, SPE, Stanford University[1], and H. A. Tchelepi, SPE, Stanford University

This paper was prepared for presentation at the 2011 SPE Reservoir Simulation Symposium held in Woodlands, Texas, U.S.A., 21-23 February 2011.

Abstract

Previous research on multiscale methods for subsurface flow aims to obtain an efficient multiscale solution to the fine-scale problems. Such multiscale solution is usually a good approximation to the fine-scale problem. However, it has been reported that the multiscale solution deteriorates for high aspect ratios and channelized structures of permeability. Moreover, the multiscale solution does not converge to the fine-scale solution unless some special techniques are used.

In this paper, we propose an efficient two-stage algebraic multiscale (TAMS) method that converges to the fine-scale solution. The TAMS method consists of two stages, namely global and local stages. In the global stage, a multiscale solution is obtained purely algebraically from the fine-scale matrix. In the local stage, a local solution is constructed from a local preconditioner such as Block ILU(0) (BILU) or Additive Schwarz (AS) method. Spectral analysis shows that the multiscale solution step captures the low-frequency spectra in the original matrix very well and when combined with a local preconditioner that represents the high-frequency spectra, the full spectra can be well approximated. Thus the TAMS is guaranteed to converge to the fine-scale solution. Moreover, the spectra of the TAMS method tend to cluster together, which is a favorable property for Krylov subspace methods to converge fast. We also show that local mass conservation can be preserved if a multiscale solution step with the finite-volume restriction operator is applied before the iterative procedure exits. This allows for the TAMS method to be used to build efficient approximate solutions for multiphase flow problems that need to solve transport equations.

We test the numerical performance of the TAMS method using several challenging large-scale problems (fine-scale grid in the magnitude of one million) with complex heterogeneous structures and high aspect ratios. Different choices in the TAMS algorithm are employed, including the Galerkin or finite-volume type of restriction operator, BILU or AS preconditioner for the second stage, and the size of blocks for BILU and AS. The performance of TAMS is comparable or superior to the state-of-the-art algebraic multigrid (AMG) preconditioner when some optimal choice in the TAMS method is adopted. Moreover, the convergence of the TAMS method is insensitive to problem sizes, and the CPU time is almost linear to problem sizes. These indicate the TAMS method is efficient and robust for large-scale problems.

Introduction

The accuracy of simulating subsurface flow and transport depends strongly on the quality of the reservoir description. The permeability of natural geologic formations typically show high variability and complex heterogeneity that spans a wide range of scales: from the pore ($\sim 10\mu\mathrm{m}$) to the geological scale ($1-10$ km). With modern reservoir characterization methods and geostatistical modeling techniques, it is possible to describe the spatial distribution of the reservoir properties in great detail. Such techniques can integrate information from different sources and different scales, such as core analysis (~ 1 cm), well logs (~ 10 cm), seismic imaging, and other geophysical and dynamic production data; as a result, the geocellular model may have $O(10^7)-O(10^8)$ grid cells. It is, however,

[1]Now with ConocoPhillips

quite prohibitive to do reservoir simulation at the geo-cellular scale, even with modern computing capabilities. Thus, there has been a significant research effort to develop scalable mathematical formulations and numerical algorithms that can deal with such large-scale problems in a robust and efficient manner.

Upscaling methods have been widely used to handle the resolution gap between geocellular and the reservoir-simulation models. In upscaling, efficient properties (e.g., porosity, permeability, and transmissibility) are computed on a coarse grid, which is amenable for more complex flow computations. In flow-based upscaling methods [12], such coarse grid properties are obtained by solving local problems based on simplified flow physics and boundary conditions. Chen et al. [8] developed a local-global upscaling method that updates the boundary conditions using global information.

Multiscale methods attempt to resolve the fine-scale information accurately without direct solution of the global fine-scale equations. Recent advances in multiscale methods show great promise in efficiently simulating complex flow processes using highly detailed reservoir models. Hou and Wu [16] proposed a multiscale finite-element method (MsFEM) that captures the fine-scale information by constructing special finite-element basis functions within each element. However, the reconstructed fine-scale velocity obtained via MsFEM is not conservative. Later, Chen and Hou [9] proposed a conservative mixed finite-element multiscale method. Another multiscale mixed finite-element method was presented by [4] and Arbogast and Bryant [5]. Numerical Green functions were used to resolve the fine-scale information, which are then coupled with coarse-scale operators to obtain the global solution. [1] proposed a modified mixed finite-element method which constructs special basis functions sensitive to the nature of elliptic problems. A multiscale finite-volume method (MsFVM) was proposed by Jenny et al. [17] for heterogeneous elliptic problems. They employed two sets of basis functions — dual and primal. The dual basis functions are identical to those of Hou and Wu [16], while the primal basis functions are obtained by solving local elliptic equations with Neumann boundary conditions calculated from the dual basis functions. A conservative fine-scale velocity field can be computed from the primal basis functions. Lunati and Jenny [24, 25] investigated compressibility and gravity effects in the multiscale finite volume method. Jenny et al. [18, 19] and Lee et al. [21] further extended the MsFVM for adaptive computation, fully implicit formulations, and black-oil simulation. Zhou and Tchelepi [35] proposed an operator-based multiscale method (OBMM), in which two multiscale operators, prolongation and restriction operators, are constructed. A coarse-scale pressure equation is then constructed by applying the prolongation and restriction operators to the fine-scale flow equations. OBMM allows for easy implementation of the MsFVM and is extensible to more complicated physics. For example, it was shown that compressibility effects can be accurately captured using OBMM with little additional cost. Lee et al. [22] further extended OBMM to multiphase transport problems. Three adaptive prolongation operators for saturation are defined adaptively to maximize efficiency. Then, both flow and transport are solved in a unified multiscale framework.

Although existing multiscale methods have shown good accuracy for a wide range of problems [19, 24, 31, 21, 35, 22, 20], it is also known that multiscale finite-volume method can suffer from large errors, especially for channelized permeability fields [23], or high anisotropy (grid spacing and permeability tensors) ratios [20]. Hajibeygi et al. [14] proposed an iterative multiscale finite-volume (i-MSFV) method in order to improve the quality of the reconstructed fine-scale solution. They showed that the multiscale solution converges to the fine-scale solution when local boundary conditions are updated with global information in each iteration. Such 'global' information was obtained by applying several line-relaxation smoothing after a multiscale solution step. The global information was then used to solve local 'correction' functions. The method can be expensive as each iteration involves several smoothing steps and requires solution of correction functions in all the local regions.

To efficiently obtain converged solutions to massive fine-scale flow problems, a wide range of linear preconditioning strategies have been developed. A preconditioner M of a matrix A is a matrix such that $M^{-1}A$ has a smaller condition number than A and hence converges faster for an iterative linear solver than the original matrix. Among the wide range of preconditioners, domain decomposition is a class of methods of particular interest mainly due to its efficiency and parallelism [29, 32]. Aarnes and Hou [2] used the multiscale finite element method as the coarse-grid solver of the non-overlapping domain decomposition preconditioner for finite-element discretization of elliptic problems. More analysis on how the condition number of the preconditioner depends on the variability in the coefficients (permeability or transmissibility) was carried by Graham et al. [13]. Both papers showed that using the multiscale finite element method as the coarse solver results in better convergence than traditional domain decomposition methods that use linear or polynomial finite element methods as coarse solvers. However, the multiscale finite element method does not possess the nice property of local mass conservation that the multiscale finite volume method has. Local mass conservation is quite important for solving nonlinear subsurface flow and transport problems. With most domain decomposition preconditioners, the numerical solution must be computed to very tight tolerances in order to ensure that the mass-balance errors are small. Otherwise, severe errors arise when the computed flow field (pressure and total-velocity) is used to solve the nonlinear transport problems. Our purpose in this paper is twofold: 1) construct an efficient multiscale preconditioner that converges to the fine-scale solution; 2) ensure a locally conservative solution at any iteration before full convergence is achieved.

We propose a flexible Two-stage Algebraic Multiscale ((TAMS) Solver. In the first stage, a multiscale solution is constructed purely algebraically from the finite-volume approximation of the fine-scale system. This step follows the operator-based multiscale method (OBMM) [35], which is an algebraic representation of the original mulitscale finite-volume scheme [17]. Here, we further extend OBMM by constructing a prolongation operator algebraically and directly from a wirebasket ordered reduced system. Spectral analysis shows that the multiscale solution step captures the low-frequency spectra of the system very well, while it misses some high-frequency components. Therefore, a second stage is required to deal with the high-frequency spectra. Local preconditioners are excellent

candidates for this second stage. We consider Block ILU(0) (BILU) and Additive Schwarz (AS) as second-stage local preconditioners. The TAMS method represents the full spectra quite well and shows excellent convergence properties. We also describe a simple process that allows for the reconstruction of a locally conservative fine-scale velocity field at any stage of computation.

The paper is organized as follows. We summarize the governing equations for subsurface flow in porous media in Section 2. Then in Section 3, we describe the multiscale finite-volume method and discuss its major limitations. In Section 4, we describe algebraic construction of the multiscale solution for finite-volume discretization of elliptic flow problems. In Section 5, we present the two-stage Algebraic Multiscale Solver (TAMS). Spectral analysis is applied to show that the two-stage method can resolve the full spectra of the fine-scale matrix and achieve fast convergence. It is also demonstrated that the TAMS method allows for the reconstruction of a locally conservative fine-scale velocity field. In Section 6, we first study the convergence and conservation properties of TAMS with two-dimensional numerical examples; then challenging three-dimensional problems with complex heterogeneous structures and high aspect ratios are presented and analyzed. The conclusions are given in the last section.

Governing Equations

Immiscible multiphase flow in porous media can be described by the mass conservation equation for each phase, i.e.,

$$\phi \frac{\partial S_l}{\partial t} + \nabla \cdot \boldsymbol{u}_l = -q_l \tag{1}$$

where the subscript l denotes a phase, ϕ is the porosity of the porous media, S_l is the saturation of phase l and $\boldsymbol{u}_l$ is the phase velocity which is related to pressure by Darcy's law:

$$\boldsymbol{u}_l = -\frac{\boldsymbol{k} k_{rl}}{\mu_l} \nabla p. \tag{2}$$

Here $\boldsymbol{k}$ is the absolute permeability of the porous media, k_{rl} is the relative permeability, which is a function of saturation, μ_l is the viscosity of phase l, and p is the pressure. For simplicity, we ignore gravity, capillary pressure, and the compressibility of fluids and rock. Hence, the porosity and viscosities are constants.

The pressure equation can be obtained by summing **Eq.(1)** for all the phases with **Eq.(2)** and the condition $\sum_l S_l = 0$:

$$\nabla \cdot (\boldsymbol{\lambda} \cdot \nabla p) = q, \tag{3}$$

where $\boldsymbol{\lambda} = \boldsymbol{k} \sum_l \frac{k_{rl}}{\mu_l}$ and $q = \sum_l q_l$. Our focus in this paper is on solving this elliptic pressure equation. Note that the absolute permeability $\boldsymbol{k}$ is generally a second-order tensor. We assume the coordinates are aligned with the principle directions of $\boldsymbol{k}$, i.e.,

$$\boldsymbol{k} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}.$$

The absolute permeability is a complex function of space and reflects the multiscale heterogeneous structure of the porous formation. Thus, usually a detailed description is required to resolve the large variability of $\boldsymbol{k}$, which in turn requires highly detailed grids (i.e., more than one million cells).

The total-velocity $\boldsymbol{u}$ is defined as the sum of all phase velocities, i.e.,

$$\boldsymbol{u} = \sum_l \boldsymbol{u}_l = -\boldsymbol{\lambda} \cdot \nabla p. \tag{4}$$

The the transport equation for each phase can be written as

$$\phi \frac{\partial S_l}{\partial t} + \nabla \cdot f_l \boldsymbol{u} = -q_l, \tag{5}$$

where q_l is fractional flow which is given by

$$f_l = \frac{k_{rl}/\mu_l}{\sum_l k_{rl}/\mu_l}. \tag{6}$$

The transport equation is a nonlinear hyperbolic PDE, and it is very important to have a locally conservative total-velocity (**Eq.(4)**) field in order to obtain accurate solutions of the saturation distribution in space and time.

Multiscale Finite-Volume Method (MsFVM)

Various multiscale methods have been proposed to solve the elliptic pressure equation [16, 1, 4, 5, 17]. The multiscale finite-volume method (MsFVM, Jenny et al. [17]) is of particular interest, mainly due to three reasons. First, the MsFVM allows for the construction of a conservative fine-scale velocity field, which is essential to solve the transport equations. Second, the finite-volume formulation is quite flexible and extensible to different physics, and so far only the MsFVM has been extended to problems with compressible multiphase flow with gravity and capillary pressure [24, 25, 35, 21]. Third, adaptive computation of coupled flow and transport have been demonstrated using MsFV [22]. Here, we briefly outline the MsFVM, and then discuss the important limitations addressed in this work.

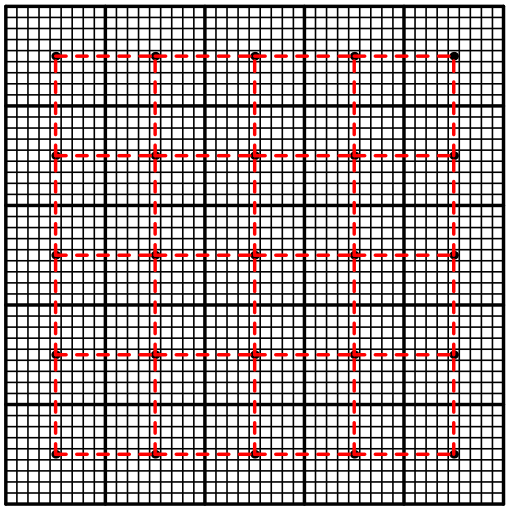


Fig. 1—A two-dimensional multiscale grid: the bold solid and dashed lines indicate the coarse grid (Ω^c) and dual coarse grid (Ω^d) respectively; solid thin lines denotes the fine grid (Ω^f).

Fig.(1) shows a simple multiscale grid in two-dimensional space. We denote the fine-scale grid as Ω^f and the coarse-scale grid, which is referred as primal coarse grid, as Ω^c. A second set of coarse grid, the dual coarse grid which is denoted by Ω^d, is constructed by connecting centers of the primal coarse grid. One cell in a grid is indicated by indexed subscript of that grid, i.e., Ω_i^f means fine cell i in the fine-scale grid. Assume the fine grid contains n_f cells and the primal coarse grid contains n_c cells. The coarsening ration Cr is defined as

$$C_r = \frac{n_f}{n_c}. \tag{7}$$

The MsFVM defines basis function on each dual cell as

$$\begin{aligned} &\nabla \cdot \left(\boldsymbol{\lambda} \cdot \nabla \Phi_I^K\right) = 0 \quad \text{in } \Omega_I^d, \\ &\frac{\partial}{\partial x_t}\left(\lambda_t \frac{\partial}{\partial x_t}\Phi_I^K\right) = 0 \quad \text{on } \partial\Omega_I^d, \end{aligned} \tag{8}$$

where the subscript t denotes the direction tangential to the boundary of the dual cell. At the vertex x_J of the dual cell, the values of the basis function should be either 0 or 1, i.e., $\Phi_I^K(x_J) = \delta_{KJ}$, where δ_{KJ} is the Kronecker delta.

With the basis functions defined in **Eq.(8)**, the fine-scale pressure p^f can be written as interpolation of coarse-scale pressure p^c (which is also the pressure at dual cell vertices):

$$p^f(x) = \sum_{K=1}^{4} \Phi_I^K(x) p_I^c \quad \text{if } x \in \Omega_I^d. \tag{9}$$

The coarse-scale pressure is obtained from a coarse-scale system, which is derived from the finite-volume discretization on the coarse-scale, i.e.,

$$\int_{\Omega_I^c} \nabla(\boldsymbol{\lambda} \cdot \nabla p)\mathrm{d}\Omega = \int_{\Omega_I^c} q\mathrm{d}\Omega. \tag{10}$$

The left hand side of **Eq.(10)** can be computed by applying **Eq.(9)** and

$$\int_{\Omega_I^c} \nabla(\boldsymbol{\lambda} \cdot \nabla p)\mathrm{d}\Omega = \oint_{\partial\Omega_I^c} \boldsymbol{n} \cdot (\boldsymbol{\lambda} \cdot \nabla p)\mathrm{d}\Gamma. \tag{11}$$

Then the coarse scale equation can be constructed as

$$\sum_{I=1}^{n_c} T_{IJ} p_I^c = q_J \quad (J = 1, \ldots, n_c), \tag{12}$$

where T_{IJ} is the so-called coarse-scale effective transmissibility, which is a function of basis functions.

The MsFVM honors mass conservation on the coarse scale according to **Eq.(10)**. However, mass conservation on the fine scale is not ensured, due to using the reduced boundary condition for the basis functions **Eq.(8)**. One has to reconstruct a conservative fine-scale velocity in order to solve the transport equation, which can be achieved by solving local elliptic problems for each primal coarse cell:

$$\begin{aligned} \boldsymbol{u} &= -\boldsymbol{\lambda}\nabla p, \quad \nabla \cdot \boldsymbol{u} = q \quad \text{in } \Omega_I^c, \\ \boldsymbol{u} &= -\boldsymbol{\lambda}\nabla p^f \quad \text{on } \partial\Omega_I^c, \end{aligned} \tag{13}$$

where p^f denotes the fine-scale pressure obtained from MsFVM (**Eq.(9)**). The velocity constructed from **Eq.(13)** is locally conservative on the fine scale [17–19].

Although the accuracy and efficiency of MsFVM have been demonstrated by a wide range of problems with strongly heterogeneous permeability and additional physical mechanisms, such as gravity, compressibility and capillarity [17–19, 24, 21, 35], there are some important limitations. One limitation has to do with channelized permeability fields, where complex channel-like structures with extremely high, or extremely low, permeability extend beyond the coarse-grid scale. One example of that is the bottom 35 layers of the SPE 10 problem [10]. The other limitation relates to strongly anisotropic transmissibility fields, due to large grid aspect ratios and highly anisotropic tensor permeability. These two limitations are strongly related to pronounced non-monotonicity of the MsFVM coarse-scale operator as transmissibility anisotropy increases. As pointed by [15], the MsFVM coarse-scale operator can be viewed as a multipoint flux approximation (MPFA) [3], and MPFA schemes are known to suffer from non-monotonicity for high grid aspect ratios and strongly anisotropic permeability fields.

Here, we discuss two examples that demonstrate these behaviors. In the first example, the bottom layer of the SPE 10 problem, which is shown in **Fig.(2a)**, is used. The reference fine grid is 220×54, and a coarse grid of 22×6 (a coarsening ratio of 90) is used in the MsFV method. The transmissibility is isotropic, i.e, $k_x = k_y$ and $\Delta_x = \Delta_y$, where Δ_x and Δ_y are the fine-cell sizes in the x and y directions, respectively. The permeability is highly heterogeneous with structures that are long and thin. These 'channelized' permeability fields characteristic of fluvial systems. We analyze the pressure equation of single-phase flow in this 2D problem. The pressure is fixed at 5000 psi and 1000 psi on the left and right boundaries; no-flow boundary conditions are specified on the top and bottom boundaries. The reference fine-scale and MsFVM pressure solutions are shown in **Fig.(2)**. Since the problem is elliptic, the pressure should be bounded by the pressure values at the boundaries, i.e., 1000 psi and 5000 psi. However, as shown in **Fig.(2c)**, the MsFV pressure is exceeds these bounds at several locations. The MsFV pressure in the [1000,5000] interval is plotted in **Fig.(2d)**. These plots indicate that the MsFVM solution captures the overall pressure distribution quite well, but suffers from pockets with very large errors at a few locations.

The second example employs a log-normal distributed permeability field. The logarithm of permeability is generated by Sequential Gaussian Simulation [11] with a dimensionless correlation length of 0.2 in each direction, as shown in **Fig.(3)**. The fine grid is 70×70 and the coarse grid used in MsFVM is 10×10. The permeability in the x and y directions are equal, while $\Delta_y = 100\Delta_x$, which leads to a highly anisotropic transmissibility. The same boundary conditions as in the previous example are used. The reference and MsFVM pressures are shown in **Fig.(3)**. It can be seen from **Fig.(3c)** that the MsFVM pressure is out of bounds (i.e., larger then 5000 psi) in some cells. **Fig.(3d)** shows the MsFVM solution in in the interval [1000,5000], and it is clear that the errors are quite large.

Algebraic Multiscale Solver(AMS)

Zhou and Tchelepi [35] presented an algebraic multiscale framework to solve **Eq.(3)**. The main idea is to cast the multiscale methodology into algebraic operations using two multiscale operators, namely restriction and prolongation. With these two operators, a multiscale method can be easily represented. In this section, we further extend the algebraic multiscale framework in Zhou and Tchelepi [35] by algebraically constructing the basis functions directly from the linear system. In this way, with information about the coarsening strategy, the multiscale method can be used essentially as a black-box linear solver.

Algebraic Multiscale Solver

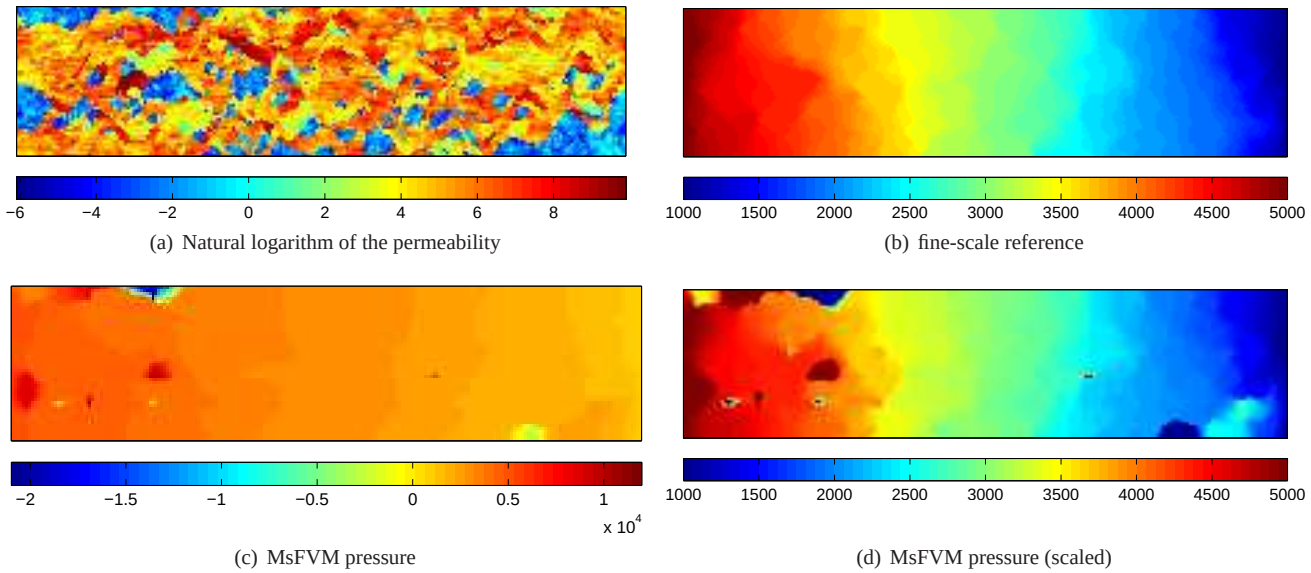
The fine-scale finite-volume discretization of **Eq.(3)** is written as

$$Ap^f = q \quad \text{for} \quad p^f \in \Omega^f, \tag{14}$$

where vectors in the space Ω^f are column vectors of dimension n_f. The direct solution of **Eq.(14)** is too expensive. In AMS, the system is projected into the coarse space by defining two multiscale operators: prolongation and restriction.

The prolongation (interpolation) operator defines the mapping from the coarse to the fine space:

$$p^f = \mathcal{P} p^c, \tag{15}$$



(a) Natural logarithm of the permeability (b) fine-scale reference

(c) MsFVM pressure (d) MsFVM pressure (scaled)

Fig. 2—Permeability and pressure solutions for the SPE 10 bottom layer

where the prolongation operator $\mathcal{P}$ is of dimension $n_f \times n_c$. Details on how to construct the prolongation operator are discussed in next subsection.

The restriction operator, $\mathcal{R}$, is describes restriction from fine to coarse spaces, and it is used to construct the coarse-grid system from the fine-grid equations. If the finite-volume approximation is used for both the fine and coarse grids, for a given coarse cell, the restriction operator is the sum of the equations of all the fine cells contained in the coarse cell. So, this finite-volume type of restriction operator can be written as

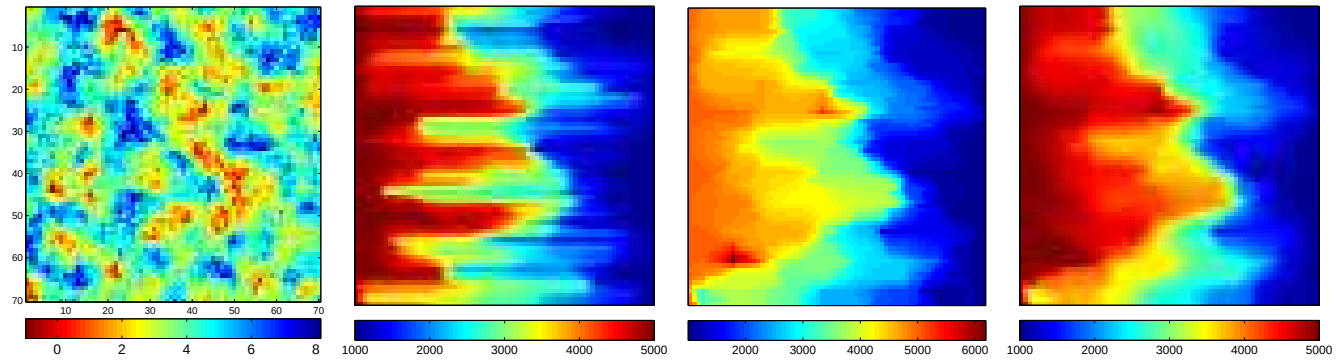
$$\mathcal{R}_{K,i} = \begin{cases} 1 & \text{if } \Omega_i^f \subset \Omega_K^c \\ 0 & \text{otherwise} \end{cases} \qquad (K = 1, \ldots, n_c;\ k = 1, \ldots, n_f). \tag{16}$$

It is also possible to use a Galerkin restriction operator as in domain decomposition methods[29, 32, 33]. The Galerkin type of restriction operator is the transpose of the prolongation operator:

$$\mathcal{R} = \mathcal{P}^T. \tag{17}$$

The coarse-grid system can be written as

$$\mathcal{R}A\mathcal{P}p^c = \mathcal{R}q. \tag{18}$$



(a) logarithm of permeability (b) fine-scale reference pressure (c) MsFVM pressure (d) MsFVM pressure (scaled)

Fig. 3—Permeability and pressure solutions for the anisotropic case

The matrix and vector associated with the coarse grid are written as

$$\begin{aligned} A^c &= \mathcal{R}A\mathcal{P}, \\ q^c &= \mathcal{R}q. \end{aligned} \tag{19}$$

The multiscale solution is then constructed as follows:

$$p^{ms} = \mathcal{P}\left(A^c\right)^{-1} q^c \equiv \mathcal{P}(\mathcal{R}A\mathcal{P})^{-1}\mathcal{R}q. \tag{20}$$

As discussed by Zhou [34], Zhou and Tchelepi [35], the algebraic multiscale method based on the finite-volume restriction operator is an algebraic representation of the original multiscale finite-volume method [17]. On the other hand, the Galerkin restriction operator leads to the multiscale finite element method [16], in which the finite-element grid correspond to the dual grid in our nomenclature here. Instead of relying on a single-pass reconstruction of the fine-scale solution, once cast in algebraic form, multiscale methods can be easily used as preconditioners as follows:

$$M_{ms}^{-1} = \mathcal{P}(\mathcal{R}A\mathcal{P})^{-1}\mathcal{R}. \tag{21}$$

Algebraic Construction of the Prolongation Operator

The prolongation operator can be assembled directly from the basis functions (**Eq.(8)**) as in existing multiscale finite-volume methods [17, 35]. However, this construction requires solution of local problems for each dual cell and needs access to all the physical quantities (e.g., mobility, grid information, etc.). Moreover, when the grid gets more complex (i.e., irregular, unstructured) the construction and assembly can get quite complicated. Here, we construct the prolongation operator directly from the fine-scale system (**Eq.(14)**). In addition to the fine-scale system, the only additional information relates to the coarse grid, which can be constructed in a preprocessing step.

Nordbottern and Bjørstad [26] discussed the relationship between the multiscale finite-volume method and non-overlapping domain decomposition preconditions that employ approximate solutions to the Schur complement of the original matrix. They showed that the multiscale finite-volume method is equivalent to the tangential component approximation [7] in the absence of source terms. In the tangential approximation, only the tangential interfacial fluxes are considered. Somewhat motivated by that work, we construct the prolongation operator from a reduced system of the original fine-scale system that incorporates the tangential approximation.

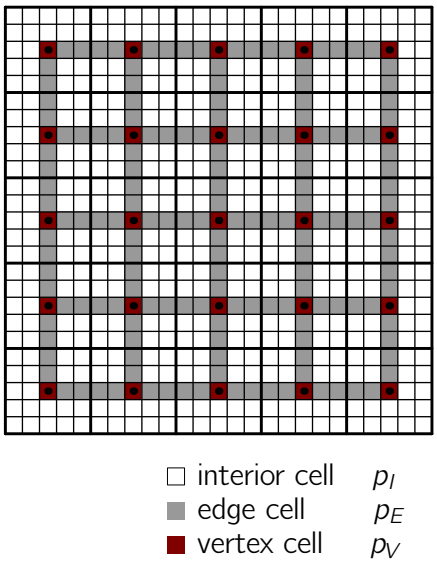


Fig. 4—The multiscale grid with the wirebasket ordering

We proceed by first introducing a wirebasket ordering of the original system [29, 33]. In two-dimensional problems with structured grid, the fine cells are grouped into three categories: interior, edge, and vertex cells as indicated in **Fig.(4)**. The categories are classified with respect to the dual grid (refer to **Fig.(1)**). In other words, vertex and edge cells are on the vertex and edge of a dual coarse-cell, respectively, while an interior cell is inside a dual coarse cell. In three-dimensional problems with structured grids, we need to introduce face cells, which are usually lumped together with edge cells. Generally unstructured grids require special treatment[33] and are beyond the scope of this paper.

We describe the method for 2D problems only. Extension to three-dimensional structured problems is straightforward, and one only need to substitute the edge cells by lumping the edge and face cells. A so-called wirebasket ordered system of the original matrix can

be written as

$$\begin{bmatrix} A_{II} & A_{IE} & 0 \\ A_{EI} & A_{EE} & A_{EV} \\ 0 & A_{VE} & A_{VV} \end{bmatrix} \begin{bmatrix} p_I \\ p_E \\ p_V \end{bmatrix} = \begin{bmatrix} q_I \\ q_E \\ q_V \end{bmatrix}. \tag{22}$$

The reduced boundary condition (**Eq.(8)**) used in MsFVM ignores flow between edge and interior cells and is equivalent to the tangential approximation in Nordbottern and Bjørstad [26]. The reduced system is constructed as follows:

$$\begin{bmatrix} A_{II} & A_{IE} & 0 \\ 0 & \tilde{A}_{EE} & A_{EV} \\ 0 & A_{VE} & A_{VV} \end{bmatrix} \begin{bmatrix} p_I \\ p_E \\ p_V \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ q_V \end{bmatrix}, \tag{23}$$

where $\tilde{A}_{EE}$ is the diagonal matrix associated with the tangential flow only, i.e., $\tilde{A}_{EE}p_E + A_{EV}p_V = 0$ describes the tangential flow equations for the edge cells. Also note that the right hand sides of the interior and edge equations are neglected in the reduced system (**Eq.(23)**), which is consistent with **Eq.(8)**. Our purpose here is not to solve the reduced system. Instead, we establish a relationship between the vertex cells and all the other cells, and that allows us to construct an algebraic prolongation operator. We solve the edge equations first, followed by the interior equations from **Eq.(23)**, i.e.,

$$\begin{aligned} p_E &= -\tilde{A}_{EE}^{-1}A_{EV}p_V, \\ p_I &= -A_{II}^{-1}A_{IE}p_E = A_{II}^{-1}A_{IE}\tilde{A}_{EE}^{-1}A_{EV}p_V, \end{aligned} \tag{24}$$

or,

$$\begin{bmatrix} p_I \\ p_E \\ p_V \end{bmatrix} = \begin{bmatrix} A_{II}^{-1}A_{IE}\tilde{A}_{EE}^{-1}A_{EV} \\ -\tilde{A}_{EE}^{-1}A_{EV} \\ I_{VV} \end{bmatrix} p_V, \tag{25}$$

where I_{VV} is an identity matrix of dimension $n_c \times n_c$. Note that a vertex cell is also the center of a primal coarse cell, and the finite-volume approximation p_V is equivalent to p^c. We denote the permutation matrix of the wirebasket ordering as G:

$$p^f = G \begin{bmatrix} p_I \\ p_E \\ p_V \end{bmatrix}. \tag{26}$$

Combining **Eq.(25)** and **(26)**, one obtains

$$p^f = G \begin{bmatrix} A_{II}^{-1}A_{IE}\tilde{A}_{EE}^{-1}A_{EV} \\ -\tilde{A}_{EE}^{-1}A_{EV} \\ I_{VV} \end{bmatrix} p^c, \tag{27}$$

which defines the prolongation operator as

$$\mathcal{P} = G \begin{bmatrix} A_{II}^{-1}A_{IE}\tilde{A}_{EE}^{-1}A_{EV} \\ -\tilde{A}_{EE}^{-1}A_{EV} \\ I_{VV} \end{bmatrix}. \tag{28}$$

Note that in **Eq.(28)**, A_{II}^{-1} and $\tilde{A}_{EE}^{-1}$ are computed for each interior region and edge independently. So, the construction of the prolongation operator based on **Eq.(28)** is local and amenable to parallel computation.

In summary, the AMS approach constructs a prolongation operator from **Eq.(28)** and then obtains the multiscale solution from **Eq.(20)**, or **(21)**. All the procedures are purely algebraic. In addition to the target fine-scale system to be solved, the approach needs information about the specific wirebasket ordering to be used.

Two-Stage Algebraic Multiscale Solver (TAMS)

The AMS scheme derived in the last section is a pure algebraic representation of the multiscale finite-volume method if the finite-volume type of restriction operator is used. In AMS, one has the flexibility to employ the Galerkin type of restriction operator, which is quite effective for problems with channelized permeability and high anisotropy ratios as we show in the next section. Here, we first investigate the convergence behavior of AMS through spectral analysis, which is then used to motivate a Two-stage Algebraic Multiscale Solver (TAMS). We also show that TAMS allows for the reconstruction of a locally conservative fine-scale velocity field after any iteration. Locally conservative solutions are essential in order to use TAMS as part of an adaptive multiscale framework for coupled flow and nonlinear transport in large-scale problems.

AMS: Convergence and Spectral Analysis

If use AMS as a preconditioner for the Richardson iterative scheme, the procedure can be written as

$$p^{\nu+1} = p^{\nu} + M_{ms}^{-1} r^{\nu}, \tag{29}$$

where ν is the iteration number, p^{ν} is the solution at iteration ν, and r^{ν} is the residual at iteration ν, which is given by

$$r^{\nu} = q - Ap^{\nu}. \tag{30}$$

The error vector, $\epsilon^{\nu} = p^{\nu} - p^{f}$, satisfies

$$\epsilon^{\nu+1} = (I - M_{ms}^{-1} A)\epsilon^{\nu}. \tag{31}$$

The iteration matrix is defined by

$$I_m \equiv I - M_{ms}^{-1} A. \tag{32}$$

The necessary sufficient condition for convergence is that the spectral radius of the AMS iteration matrix defined in **Eq.(32)** should be less than 1.

Lemma 1. *The iteration matrix I_m of AMS has eigenvalue 1 of multiplicity no less than $n_f - n_c$.*

Proof. We know that for general matrices C and D

$$\mathrm{rank}(CD) \leq \min(\mathrm{rank}(C), \mathrm{rank}(D)).$$

From the construction of the restriction and prolongation operators, $\mathrm{rank}(\mathcal{P}) = \mathrm{rank}(\mathcal{R}) = n_c$. Therefore, based on **Eq.(21)**, we conclude that

$$\mathrm{rank}(M_{ms}^{-1}) \leq n_c.$$

Using the same argument

$$\mathrm{rank}(M^{-1}A) \leq n_c,$$

which is equivalent as

$$\dim \mathrm{Ker}(M^{-1}A) \geq n_f - n_c,$$

where $\mathrm{Ker}(\cdot)$ denotes the kernel, or null space, of a matrix. We know $\forall x \in \mathrm{Ker}(M^{-1}A)$, x is also the eigenvector of $M^{-1}A$ corresponding to eigenvalue 0. Therefore, $M^{-1}A$ has eigenvalue 0 of multiplicity not less than $n_f - n_c$. In other words, I_m has eigenvalue 1 of multiplicity not less than $n_f - n_c$. □

Based on **Lemma (1)**, we see that AMS does not ensure convergence as a general linear preconditioner, due to the null space of M_{ms}^{-1}. The convergence properties of AMS can also be understood from **Lemma (2)**.

Lemma 2. *According to* **Eq.(29)**, *$p^{\nu+1} = p^{\nu}$ holds for $\nu \geq 1$ and any given initial pressure $p^{(0)}$.*

Proof. Since $M_{ms}^{-1} = \mathcal{P}A^{c}\mathcal{R}$, from **Eq.(29)**, if we show $\mathcal{R}r^{\nu} = 0$ for $\nu = 1$, it follows that $p^{\nu+1} = p^{\nu}$ for $\nu \geq 1$. We start from

$$r^{(1)} = f - Ap^{(1)} = (f - Ap^{(0)}) - AM_{ms}^{-1}(f - Ap^{(0)}). \tag{33}$$

Note that

$$\mathcal{R}AM_{ms}^{-1} = \mathcal{R}A\left(\mathcal{P}\left(A^{c}\right)^{-1}\mathcal{R}\right) = \left(\mathcal{R}A\mathcal{P}\right)\left(A^{c}\right)^{-1}\mathcal{R} = \mathcal{R}. \tag{34}$$

Thus, applying $\mathcal{R}$ on both sides of **Eq.(33)** gives

$$\mathcal{R}r^{(1)} = 0, \tag{35}$$

which leads to $p^{\nu+1} = p^{\nu}$ for $\nu \geq 1$. □

Lemma (2) shows that the Richardson iterative procedure with the AMS preconditioner should stop after one iteration, since additional iterations will not change the residual. Usually, a single-pass of AMS, which is equivalent to the original MsFVM when the finite-volume restriction operator is used, is adequate. However, if a very tight tolerance is specified, or if the problem is highly anisotropic, an iterative AMS strategy is necessary.

We now show an example of a simple two-dimensional flow problem for **Eq.(3)**. The fine grid is 40×40 and the coarse grid is 4×4, i.e., the coarsening ratio is 10×10. The permeability is isotropic and homogeneous: $k_x = k_y = 1$ and no-flow boundary conditions are specified. Pressure is fixed in cells (1, 2) and (40, 40) with values of 1 and 0, respectively. The finite-volume type of restriction operator is used to construct AMS. The spectra of the iteration matrix are shown in **Fig.(5)**. As the figure indicates clearly, the eigenvalues of the AMS iteration matrix cluster around 0 and 1. Actually, the eigenvalue 1 is of multiplicity $n_f - n_c$, which is caused by the rank

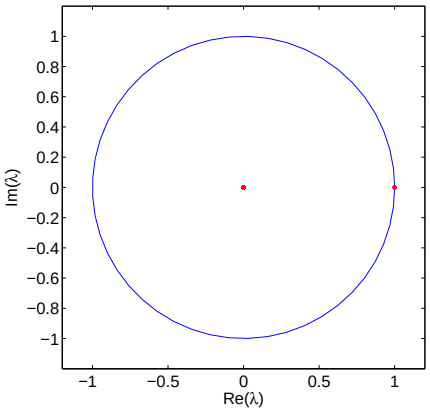
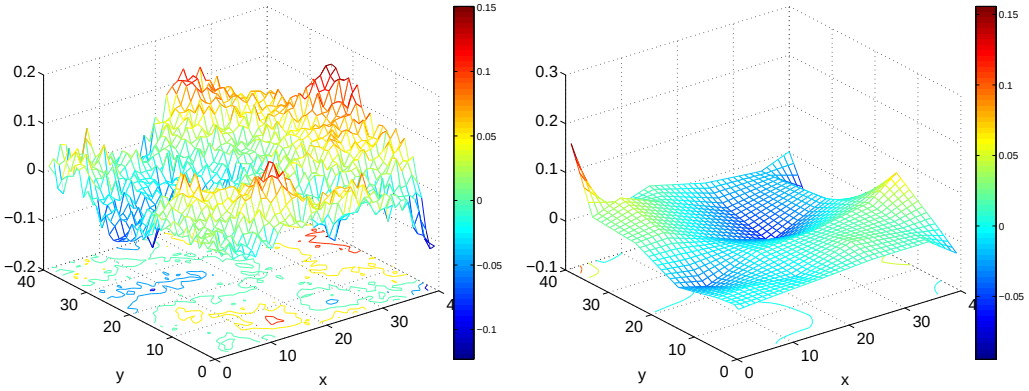


Fig. 5—The full spectra of the iteration matrix of AMS (denoted by dots) for an isotropic homogeneous case.



(a) $\lambda_{max}(I_m) = 1$ (b) $\lambda_{min}(I_m) = 0$

Fig. 6—Eigenvectors of AMS iteration matrix that are associated with the largest and smallest eigenvalues (1 and 0, respectively).

deficiency of M_{ms}^{-1} as shown in **Lemma (1)** and associated proof. However, it is quite encouraging that all the other eignevalues are clustered around 0, which means that AMS represents the remaining part of the spectrum very well.

The eigenfunctions of the AMS iteration matrix associated with the largest and smallest eigenvalues (1 and 0, respectively) are plotted in **Fig.(6)**. The missing spectra (associated with eigenvalue 1) represent high-frequency errors on the scale of a fine-grid cell. On the other hand, the AMS method perfectly resolves the low-frequency errors.

Fig.(7) and **Fig.(8)** compare the highest- and lowest-frequency eigenfunctions of A^{-1} and M_{ms}^{-1}. These figures confirm that the AMS preconditioner represents the low frequency spectra of the fine-scale matrix very well, while the high frequency components are not captured well at all.

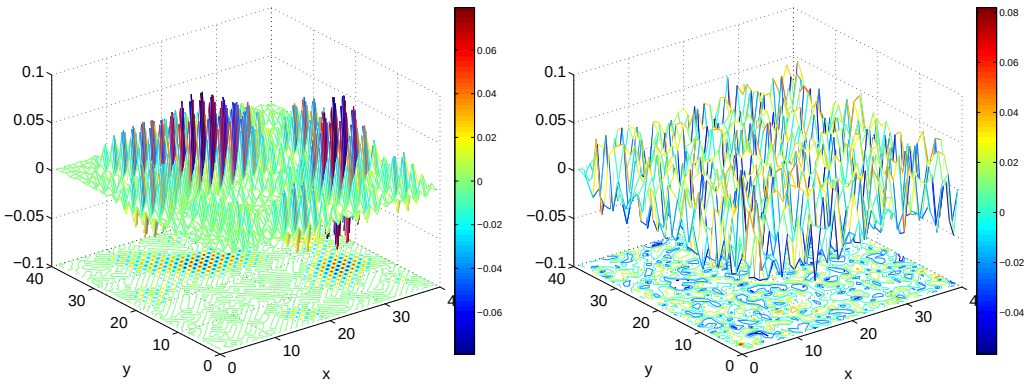
TAMS

The convergence and spectral analysis presented above clearly reveal that the AMS operator captures the low-frequency errors extremely well, but does not guarantee converge as a preconditioner for the Richardson iterative scheme, and that the crux of the challenge is that AMS cannot resolve the high frequency errors. Therefore, we propose to combine the AMS operator with a local preconditioner that can resolve the high frequency errors. We refer this two-stage approach as two-stage algebraic multiscale solver (TAMS). The TAMS algorithm is described in **Table (1)**, where M_{local}^{-1} denotes a local preconditioner. Algebraically, the TAMS preconditioner can be written as

$$M_{TAMS}^{-1} = M_{ms}^{-1} + M_{local}^{-1} - M_{local}^{-1} A M_{ms}^{-1}. \tag{36}$$

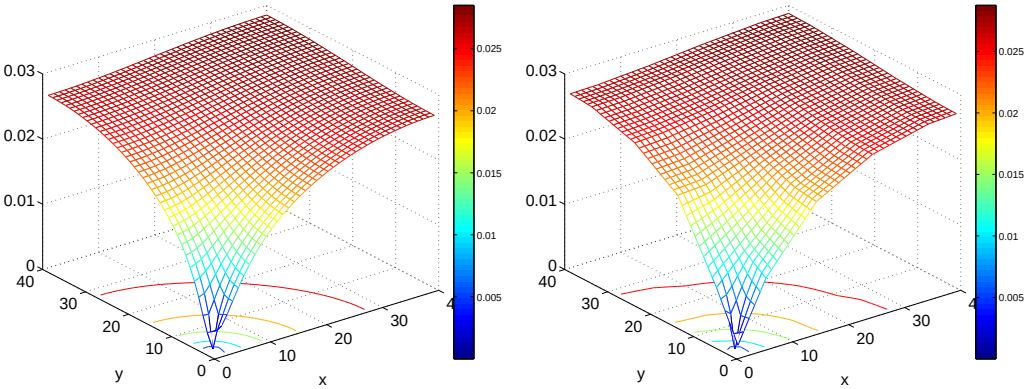
As for the specific local preconditioner to be used, block preconditioners are preferred because they are usually more suitable for parallel computation, and they resolve the local high frequency errors very well. Block ILU(0) (BILU) and the one-level additive Schwarz (AS) preconditioners are two candidates considered in this paper due to their computational efficiency.

We now take the same example as in the previous subsection. The local (second-stage) preconditioner is BILU, which has blocks



(a) A^{-1} (b) M_{ms}^{-1}

Fig. 7—Eigenvectors of A^{-1} and M_{ms}^{-1} with the highest frequency.



(a) A^{-1} (b) M_{ms}^{-1}

Fig. 8—Eigenvectors of A^{-1} and M_{ms}^{-1} with the lowest frequency.

1: $\nu = 0$
2: initialize p^{ν}
3: construct the restriction operator $\mathcal{R}$ from **Eq.(16)** or **(17)**
4: construct the prolongation operator $\mathcal{P}$ from **Eq.(28)**
5: construct the AMS preconditioner from **Eq.(21)**
6: **while** ($\nu <$ maximum iteration number && not converged) **do**
7: update with AMS:

$$p^{\nu+1/2} = p^{\nu} + M_{ms}^{-1}(q - Ap^{\nu})$$

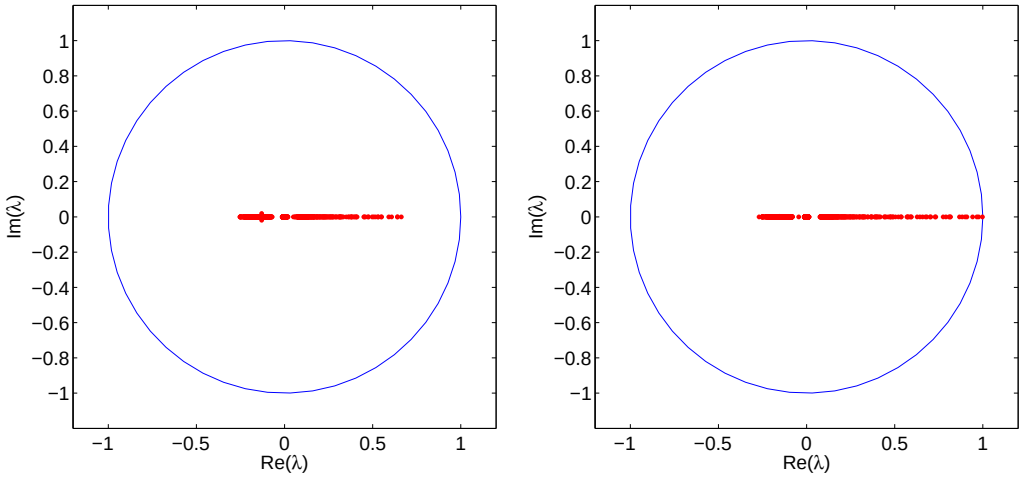
8: update with a local preconditioner:

$$p^{\nu+1} = p^{\nu+1/2} + M_{local}^{-1}(q - Ap^{\nu+1/2})$$

9: **end while**

Table 1—The TAMS algorithm for the Richardson iterative scheme.

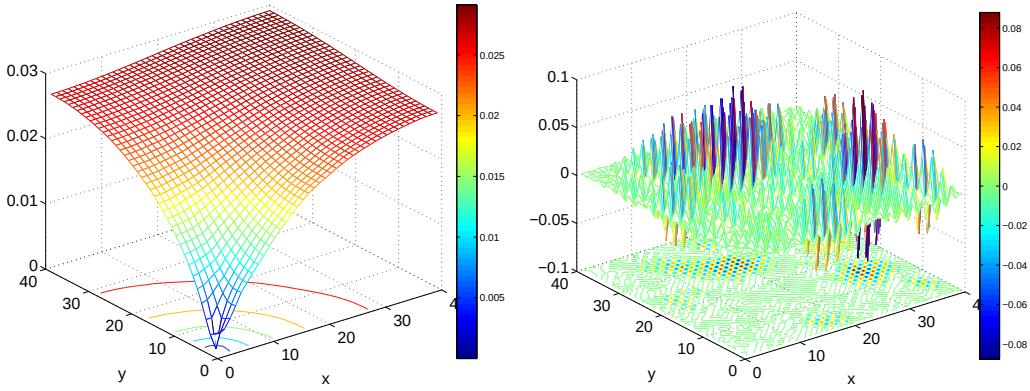
of the same size 2×2. **Fig.(9)** shows the spectra of the TAMS and BILU iteration matrices. All the eigenvalues of the TAMS iteration matrix are inside the unit circle, which means the TAMS preconditioner will converge for the Richardson iterative scheme. In contrast, the spectral radius of the BILU iteration matrix is very close to 1, which indicates that if used alone, the BILU preconditioner would converge quite slowly compared with the TAMS preconditioner. Recall that the AMS preconditioner would not converge. It is the combination of AMS with BILU that leads to a scheme that is both convergent and efficient.



(a) TAMS (b) BILU

Fig. 9—The full spectra of the iteration matrix(denoted by dots) for the isotropic homogeneous case.

Comparison between **Fig.(10)**, which shows the lowest and highest frequency eigenvectors of M_{TAMS}^{-1} for TAMS with **Fig.(7)** and **Fig.(8)**, which are for the fine-scale operator A^{-1}, it is clear that TAMS represents both the lowest and highest frequencies quite well. We also show the eigenvectors of the BILU local preconditioner (with a block size of 2×2) in **Fig.(11)**, which indicates the BILU preconditioner represents the high-frequency spectra very well and the low-frequency spectra quite poorly. The combination of AMS and BILU, i.e., the TAMS scheme, yields a very good representation of the entire spectra of the target fine-scale matrix.

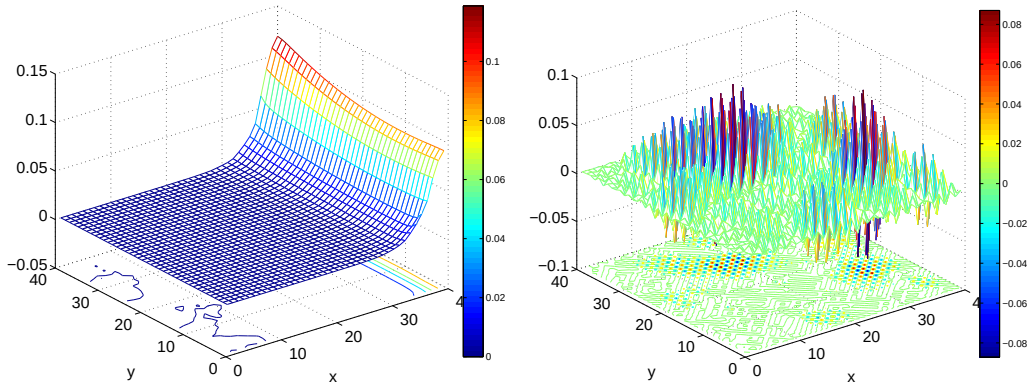


(a) the lowest frequency (b) the highest frequency

Fig. 10—Eigenvectors of M_{TAMS}^{-1} with the highest and lowest frequencies

When the size of the BILU blocks is increased, the spectral radius of the TAMS iteration matrix is reduced further and more eigenvalues are clustered around 0, as shown in **Fig.(12)** and **Fig.(13)** where the block size of BILU is increased to 4×4. However, the computational cost of BILU increases with block size.

In realistic cases of subsurface flow, the permeability shows large heterogeneity and complex correlation structures. The spectrum for a heterogeneous case, where the permeability is the top layer of the SPE10 comparative case [10], is shown in **Fig.(14)**. The permeability varies by six orders of magnitude and the variance of its natural logarithm is 5.5. The fine and coarse grids are 220×60 and 22×6. No-flow boundary conditions are applied and the pressure is fixed at fine cells (1, 0) and (80, 60) with values of 1 and 0,



(a) the lowest frequency (b) the highest frequency

Fig. 11—Eigenvectors of the local preconditioner M_{BILU}^{-1} with the highest and lowest frequencies

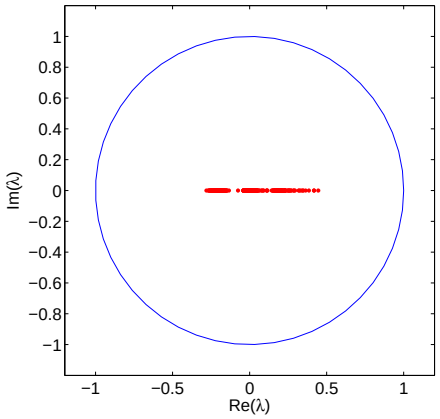


Fig. 12—The full spectra of the iteration matrix(denoted by dots) for the isotropic homogeneous case: the block size of BILU is 4×4.

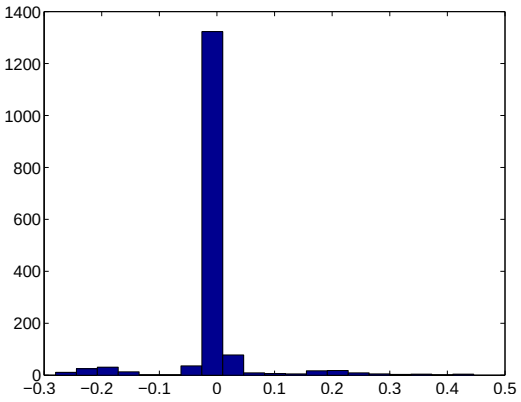


Fig. 13—The histogram of the spectra of the iteration matrix for the isotropic homogeneous case: the block size of BILU is 4×4.

respectively. The BILU block size is chosen to be 4×4. The spectra of the TAMS iteration matrix are shown in **Fig.(15)**, which clearly indicates that the TAMS preconditioner converges quickly for this highly heterogeneous case.

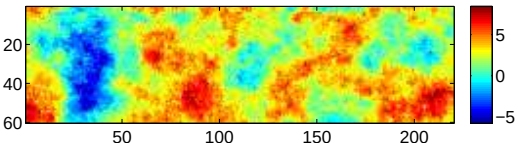


Fig. 14—The natural logarithm of the top layer of the SPE 10 permeability.

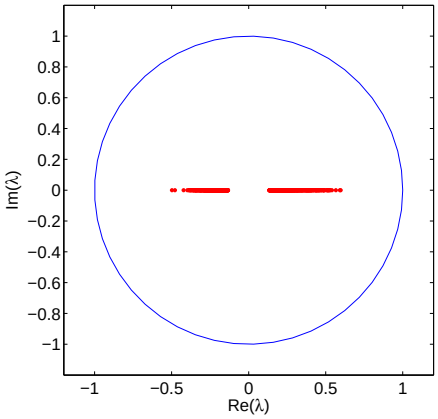


Fig. 15—The spectra of the TAMS iteration matrix for the SPE 10 top layer permeability field.

We note that Hajibeygi et al. [14] developed an iterative multiscale finite volume (i-MSFV) method, which was also shown to converge to the fine-scale solution. They constructed local correction functions on the dual coarse cells to improve local errors on the dual grid. The correction functions were updated in each iteration with global information. Line relaxation in each coordinate direction was employed as a smoother to provide the global information. The smoothing steps had to be applied several times for each iteration to ensure convergence, and the number of smoothing steps is problem dependent. This approach is not in an algebraic form and is not applied as an general linear preconditioner to combine with Krylov subspace methods, for example. The i-MSFV method employs additional global and local information from the smoother and the correction functions to improve the convergence of the original MsFVM.

TAMS: Local Mass Conservation

If the TAMS procedure in **Table (1)** stops when a target small (tight) convergence tolerance is achieved, we do not need to worry about mass conservation, since the fine-scale equation is satisfied with the desired accuracy everywhere. However, in many circumstances, we would like to maximize computational efficiency but still maintain reasonable accuracy. In practice, one may choose a relatively large (loose) convergence tolerance and a small maximum iteration number as exit conditions of the TAMS procedure. Such a strategy involves a great risk if the computed numerical approximation is not locally mass conservative. The violation of mass conservation often results in non-physical and unbounded saturations when solving the nonlinear transport equations.

Actually in AMS, except for the algebraic construction procedure, the prolongation operator defined by **Eq.(28)** is identical to that assembled from the basis functions solved from **Eq.(9)** as in Zhou and Tchelepi [35]. If the finite-volume type of restriction operator is used, the resulting multiscale solution is identical with the multiscale finite-volume method [35]. A nice property of the multiscale finite-volume method is that mass conservation is guaranteed on the coarse scale, which allows for the reconstruction of a locally conservative fine-scale velocity field as given by **Eq.(13)**. Here, we show that TAMS guarantees local mass conservation on the coarse scale, if a single-pass of AMS with the finite-volume restriction operator is used as the last step. Thus, one can also reconstruct the locally conservative velocity from the TAMS pressure according to **Eq.(13)**.

Lemma 3. *Local mass conservation is satisfied on the coarse grid if an AMS step with the finite-volume restriction operator is applied after any iteration of the TAMS algorithm in* **Table (1)**.

Proof. Denote the solution as p^ν, then $r = q - Ap^\nu$. Applying an AMS step, one obtains

$$\tilde{p} = p^\nu + \mathcal{P}(\mathcal{R}A\mathcal{P})^{-1}\mathcal{R}r. \tag{37}$$

Multiply $\mathcal{R}A$ on both side of **Eq.(37)**:

$$\mathcal{R}A\tilde{p} = \mathcal{R}Ap^\nu + (\mathcal{R}A\mathcal{P})(\mathcal{R}A\mathcal{P})^{-1}\mathcal{R}r = \mathcal{R}Ap^\nu + \mathcal{R}r = \mathcal{R}q. \tag{38}$$

Equivalently, we have

$$\mathcal{R}(q - A\tilde{p}) = 0 \tag{39}$$

Note that $\mathcal{R}$ is a finite-volume restriction operator defined in **Eq.(16)**. **Eq.(39)** shows that the coarse-grid finite-volume discretization equations are exactly satisfied everywhere, i.e., mass conservation is guaranteed for each coarse cell. □

As shown in the next section, using the Galerkin type restriction operator in TAMS is usually more robust and efficient, especially for highly anisotropic problems. However, the Galerkin type of restriction operator does not ensure mass conservation. From **Lemma (3)**, we can apply an AMS step with the finite-volume type of restriction operator at the last 'clean-up' step, which fixes the conservation problem.

In summary, the TAMS scheme offers three advantages when used as a linear-solution preconditioner : 1) small spectral radius that ensures convergence; 2) clustering of eigenvalues that allows for Krylov subspace methods to converge faster; 3) local mass conservation that permits relaxed (relatively loose) convergence tolerance when solving the pressure equation.

TAMS Numerical Performance

As discussed in the previous section, although TAMS converges for the Richardson iteration, more sophisticated Krylov subspace methods can take advantage of the clustering of the eigenvalues of TAMS and obtain faster convergence. Here, we use GMRES [28] as the Krylov solver.

Convergence Study

For our convergence study, we take the same test case as in Hajibeygi et al. [14]. A fine grid of 44×44 and a coarse grid of 4×4 are used, i.e., the coarsening ratio is 11×11. The aspect ratio α is defined as the ratio of the horizontal and to vertical grid sizes, i.e.,

$$\alpha = \frac{\Delta x}{\Delta y}. \tag{40}$$

In terms of the anisotropy transmissibility, a case with isotropic permeability and aspect ratio α is identical to one with $\Delta x = \Delta y$ and $k_x = \alpha^2 k_y$.

Both homogeneous and heterogeneous permeability with different aspect ratios, α, are considered. In the homogeneous case, $k_x = k_y = 1$, and there is a source with $q = 1/(\Delta x \Delta y)$ and a sink with $q = -1/(\Delta x \Delta y)$ in fine cells (13, 13) and (32, 32), respectively. In the heterogeneous case, the permeability is shown in **Fig.(16)**, which is part of the top layer of the SPE 10 comparative case [10]. The natural logarithm of the permeability has a variance of 6.66 and a mean of -0.29.

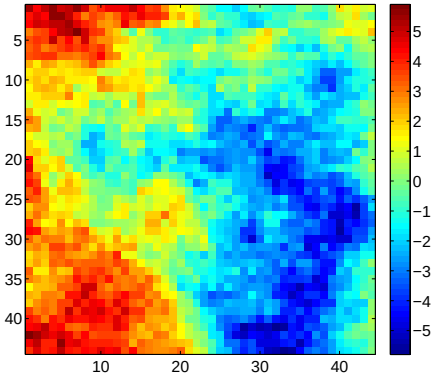


Fig. 16—The natural logarithm of heterogeneous permeability for convergence study.

In the TAMS algorithm, the Galerkin type restriction operator is used for the AMS preconditioner; BILU is chosen as the local preconditioner. The block size of BILU is fixed at 16 while two block shapes are used: one is a square block (i.e., a size of 4×4) and the other is a line block (a size of 16×1). For comparison purpose, the BILU preconditioner alone is also used. Both GMRES and Richardson solvers are considered. The error is measured by

$$\epsilon = \frac{\|p - p_f\|_\infty}{\|p_f\|_\infty}. \tag{41}$$

Fig.(17) shows the convergence histories for the test cases. BILU converges very slowly for the Richardson method, while TAMS has a very good convergence rate. GMRES improves the convergence rate further. TAMS with GMRES has good convergence for all the test cases. Compared with the iterative multiscale finite volume (i-MSFV) method [14], TAMS converges faster in the heterogeneous case especially for $\alpha = 10$. The i-MSFV method appears to be quite sensitive to both the heterogeneity level and aspect ratio. TAMS is weakly sensitive to the heterogeneity and slightly sensitive to the aspect ratio.

Conservation Study

We have shown that application of AMS with the finite-volume type of restriction operator as the last step ensures mass conservation on the coarse scale. Then, a reconstruction procedure, **Eq.(13)**, can be used to obtain a locally conservative fine-scale velocity.

T demonstrate these ideas, we consider a two-phase flow problem with the permeability from the SPE 10 bottom layer as shown in **Fig.(2a)**. The main objective is to test the mass conservation of the method. Hence, we take linear relative permeability curves, i.e. $k_{rl} = S_l$, and unit viscosity ratio, $\mu_1 = \mu_2$. No-flow boundary conditions are specified for all the boundaries of the domain. We introduce two wells in fine cells (6, 28) and (215, 28), and we use the Peaceman model [27] to describe the well equations, i.e.,

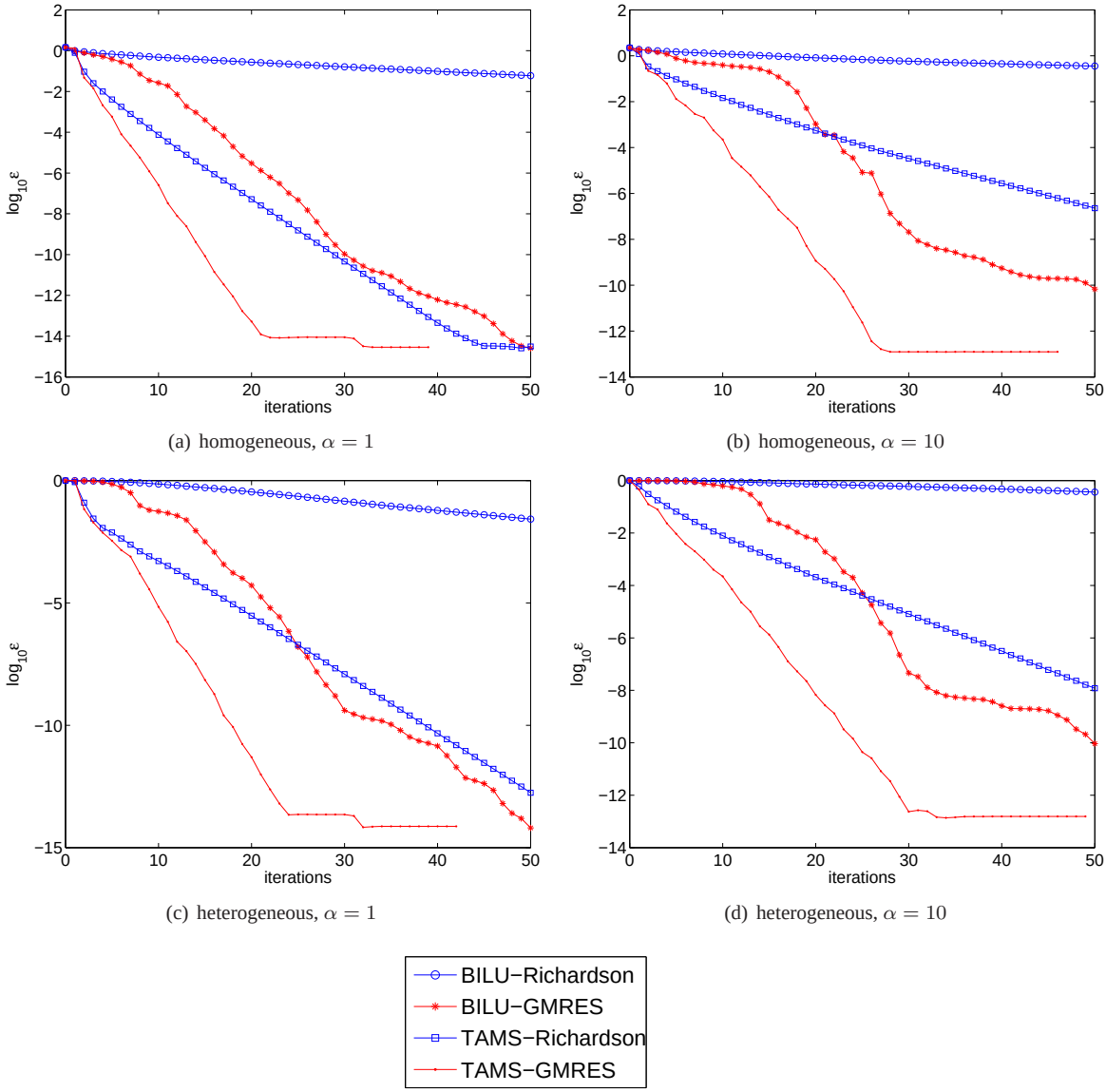
$$q^w_{l,i} = WI \lambda_{l,i}(p_i - p^w_i). \tag{42}$$

Eq.(42) defines the flow rate of phase l flowing from a reservoir cell i into the well, where p_i is the pressure in cell i, p^w_i is the pressure of the well in that cell, and WI is the well index, which is given by

$$WI = \frac{2\pi kh}{\ln \frac{r_o}{r_w}}, \tag{43}$$

where h is the length of the well. In this test case, we employ pressure control for the wells, i.e., the well pressure p^w is fixed in the two wells. We take $p^w_1 = 5000$ psi for the well at (6, 28) and $p^w_2 = 1000$ psi for the other well at (215, 28). We report results in terms of the following dimensionless time

$$\tau = \frac{\phi L_x L_y \mu}{\bar{k}(p^w_1 - p^w_2)} \tag{44}$$



(a) homogeneous, $\alpha = 1$

(b) homogeneous, $\alpha = 10$

(c) heterogeneous, $\alpha = 1$

(d) heterogeneous, $\alpha = 10$

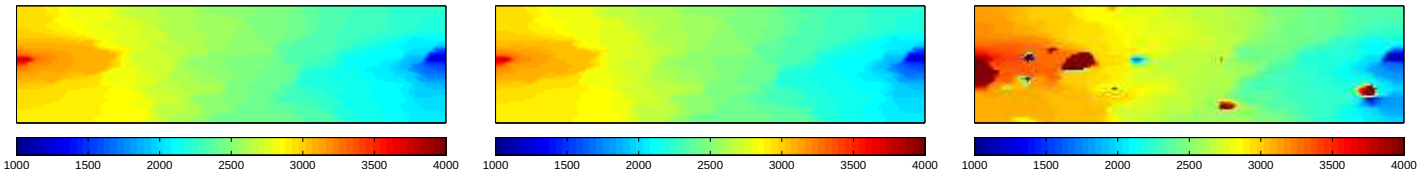
(e) legend for plots in (a,b,c,d)

Fig. 17—The convergence histories for homogeneous and heterogeneous cases with $\alpha = 1, 10$ (the legend of the plots shown in (e)).

We choose the Galerkin type of restriction operator and BILU as the local preconditioner for TAMS. The block size of BILU is 11×5. After one iteration of TAMS, the pressure error ϵ as computed from **Eq.(41)** is 3.04×10^{-2}. We compare the pressure solutions of TAMS with the fine-scale reference, as well as, AMS in **Fig.(18)**. It is clear that AMS suffers from large errors, while TAMS yields accurate results in just one iteration. However, the velocity field computed directly from the TAMS reconstructed fine-scale pressure does not ensure mass conservation. We define the mass conservation error in each fine cell Ω_i^f as

$$\epsilon_{con,i} = \frac{\| \oint_{\partial\Omega_i^f} \boldsymbol{u} \cdot \boldsymbol{n} \mathrm{d}\Gamma + q \|}{q_{inj}}, \tag{45}$$

and the conservation error after one iteration of TAMS is shown in **Fig.(19)**, where we plot the error in the interval [0, 0.05]. The largest conservation error is 0.12. The pattern of conservation errors is nearly aligned with the dual coarse grid, while it is also somewhat affected by the BILU preconditioner. The conservation errors will usually ruin the solution of the transport equation, i.e., result in unphysical and unbounded solutions.



(a) fine-scale reference (b) TAMS (c) AMS

Fig. 18—Pressure solutions of the fine-scale reference, the TAMS method (after one iteration) and the AMS method in the case of conservation study

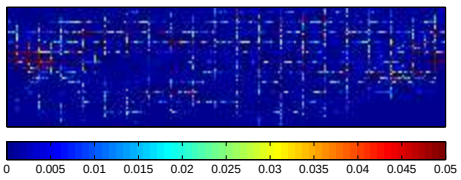


Fig. 19—Conservation errors after one iteration of TAMS

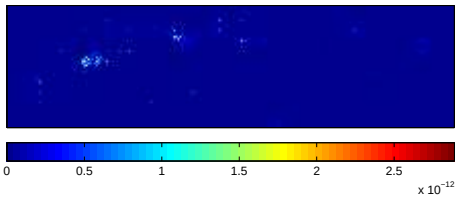


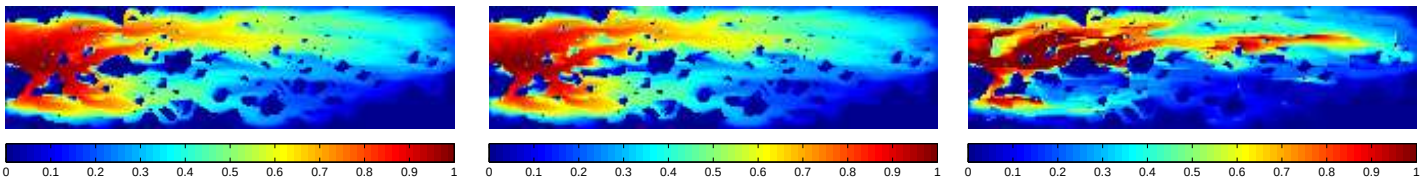
Fig. 20—Conservation errors after one iteration of TAMS followed by applying one step of AMS with the finite-volume type of restriction operator and a reconstruction process

To ensure mass conservation, after one iteration of TAMS we apply an AMS step with the finite-volume type of restriction operator and then reconstruct a conservative fine-scale velocity from **Eq.(13)**. The conservation error from this treatment is shown in **Fig.(20)**, which is very small everywhere. In **Fig.(21)**, we compare the saturation result computed using the reconstructed velocity with the fine-scale reference saturation and the saturation computed after one iteration of TAMS without any reconstruction process. The saturation from the reconstructed velocity is close to the fine-scale reference, while the saturation directly from TAMS show much larger error. Actually, the saturation directly from TAMS is not bounded in [0, 1], and **Fig.(21c)** is plotted in the scale of [0, 1] (i.e., out-of-bound values are truncated to the nearest bound). The maximum value is 1.24×10^3, which indicates the saturation solution suffers a lot from mass conservation errors.

We measure the saturation error by the L-2 norm, i.e.,

$$e_s = \|S - S_{ref}\|_2. \tag{46}$$

Fig.(22) shows the saturation errors for different iteration steps, which are all followed by the reconstruction process. It is clear that the saturation error is decreases with the iteration.



(a) fine-scale reference saturation (b) saturation from reconstructed velocity (c) saturation directly from TAMS (one iteration)

Fig. 21—Fine-scale reference saturation and the saturation computed from the reconstructed velocity or directly from the TAMS

Efficiency Study

The TAMS method is proposed to solve large-scale problems with high permeability variability and complex correlation structures. Two large three-dimensional cases are studied here. The first one has a log-normally distributed permeability with $128 \times 128 \times 64$ fine cells. The natural-logarithm of permeability is generated by sequential Gaussian simulation Deutsch and Journel [11] from a Gaussian distribution with mean and variance of 4.0. The correlation lengths are one tenth of the physical dimensions. The permeability is shown in **Fig.(23)**, where the permeability values vary by five orders of magnitudes. The second example is the 10th SPE comparative solution model [10] ,which is known for its highly detailed complex geological description. It contains $220 \times 60 \times 85$ fine cells (1.1 million).

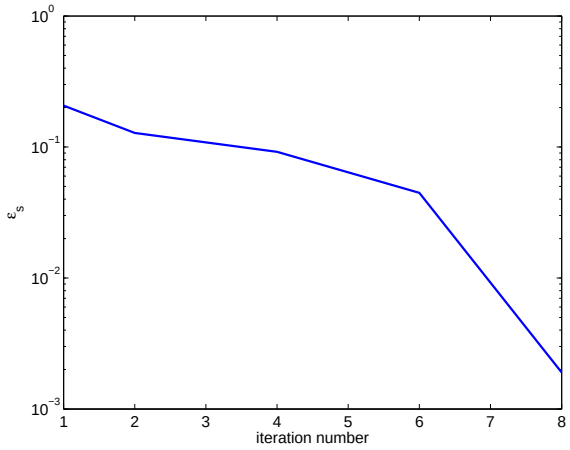


Fig. 22—Saturation error versus the number of TAMS iterations

The top 35 layers are of Tarbert formation and represent a prograding near-shore environment, while the bottom 50 layers are fluvial and contain lots of channel-like structures.

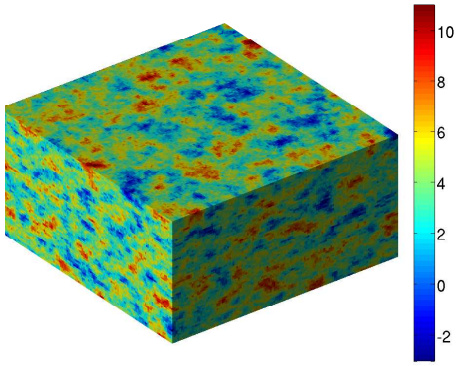


Fig. 23—The natural logarithm of the log-normal distributed permeability field.

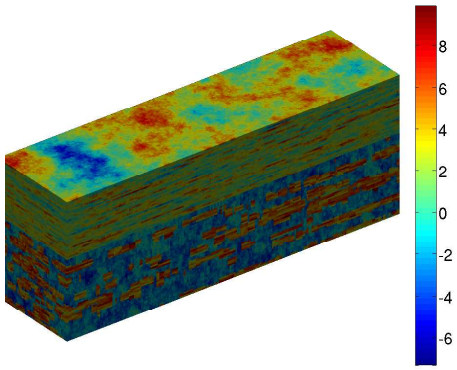


Fig. 24—The logarithm of the SPE 10 permeability field.

In both cases, Dirichlet boundary conditions are applied on the left and right faces with normalized pressures of 1 and 0, respectively; no-flow boundary conditions are specified for all other boundaries. Isotropic and anisotropic permeability fields are studied . In the isotropic problem, $\Delta x = \Delta y = \Delta z$ and $k_x = k_y = k_z$. In the anisotropic one, $\Delta x = \Delta y = \Delta z$ and $k_y = 1000k_x, k_z = 10k_x$. In the two cases, k_x is given by the permeability shown in **Fig.(23)** and **(24)**, respectively.

The coarse grid is $16 \times 16 \times 8$ for the log-normal case (coarsening ratio $8 \times 8 \times 8$) and $22 \times 6 \times 17$ for the SPE 10 case (coarsening ratio $10 \times 10 \times 5$. Solutions given by TAMS are shown in **Fig.(25)** and **(26)** for the two cases, respectively. GMRES is used as the outer (Krylov) method, and the stopping criterion is a reduction in the relative L-2 norm of the residual by five-orders of magnitude i.e., $\|r_k\|_2/\|r_0\|_2 \leq 10^{-5}$.

Our objective here is to investigate the efficiency of TAMS and evaluate various options within the TAMS algorithm. As described earlier, in TAMS one can choose a Galerkin, or a finite-volume, type of restriction operator. There are also numerous options for the local preconditioner. Here, we limit ourselves to BILU and Additive Schwarz (AS). There is also a variety of choices for the parameters of the two preconditioners, for example, the dimension of the BILU blocks, the size of the overlapping region in AS. Two different dimensions are considered for the BILU blocks: $2 \times 2 \times 4$ and $4 \times 4 \times 4$ in the log-normal case; $2 \times 2 \times 5$ and $4 \times 4 \times 5$ in the SPE 10 case. For AS, we choose the blocks to be the same as the primal coarse cells and overlapping of 0 and 1 in each dimension are considered. **Table (2)** lists the options and parameters investigated here.

Table (3) shows the GMRES iteration numbers and the linear-solver CPU time for the two cases. The code is written in C++ and built around the PETSC [6] linear libraries. We have not done any performance tuning. It is quite clear from **Table (3)** that the Galerkin restriction operator is superior to the finite-volume type restriction operator. Also, BILU is more efficient than AS. The combination of the Galerkin restriction operator and BILU yields the most efficient and robust results in all the test cases. As for BILU, larger block

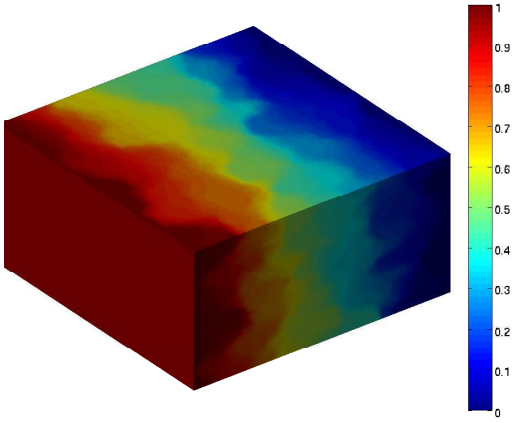


Fig. 25—Solution of the log-normal case.

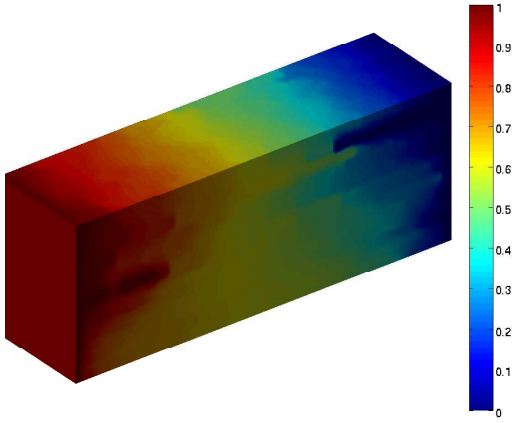


Fig. 26—Solution of the SPE 10 case.

restriction operator of AMS	FVM or Galerkin
local preconditioner	BILU or ASM
Dimension of BILU blocks	$2 \times 2 \times 4(16)$ or $4 \times 4 \times 4(64)$ for log-normal $2 \times 2 \times 5(20)$ or $4 \times 4 \times 5(80)$ for SPE 10
Overlapping of AS blocks	0 or 1

Table 2—Various options and parameters for TAMS

sizes have a better convergence rate, but then each iteration becomes more expensive. An optimal block-size would depend on the computer architecture. For AS, the overlapping of local blocks barely increases the convergence rate, while the CPU cost gets much more expensive. Thus, overlapping AS is not recommended as the local preconditioner.

				log-normal				SPE 10			
				Isotropic		Anisotropic		Isotropic		Anisotropic	
				its.	cpu	its.	cpu	its.	cpu	its.	cpu
Galerkin	BILU	block dimen.	16; 20	13	8.1	30	19.2	22	14.5	43	26.5
			64; 80	11	8.4	28	22.1	16	14.8	37	29.1
	AS	overlapping	0	20	16.2	86	48.5	34	27.5	> 100	-
			1	19	26.2	84	76.2	28	36.7	> 100	-
FVM	BILU	block dimen.	16; 20	14	8.1	30	21.0	> 100	-	> 100	-
			64; 80	11	8.3	32	23.9	> 100	-	> 100	-
	AS	overlapping	0	21	15.8	> 100	-	> 100	-	> 100	-
			1	19	25.9	> 100	-	> 100	-	> 100	-

Table 3—GMRES iterations and solver CPU time (seconds) of TAMS with different options and parameters for the two three-dimensional cases.

The best options for the TAMS algorithm are the Galerkin type of restriction operator and the BILU preconditioner. Using these options, the iteration numbers in the anisotropic cases are slightly more than double those for the the corresponding isotropic cases. Recall that the transmissibility anisotropy is as high as 1000 in the $x - y$ plane, and 100 in the $y - z$ plane. Notice that TAMS with the finite-volume type of restriction operator fails to convergence within 100 iterations in the SPE10 case. As discussed earlier, MsFVM robustness and performance deteriorate for channelized permeability fields, in which the coarse-scale MsFVM suffers from non-monotonicity. The Galerkin type of restriction operator, which corresponds to the multiscale finite-element method, is more robust and efficient for such difficult problems.

We also show a comparison of TAMS with what is considered as the state-of-the-art preconditioner: the algebraic multigrid (AMG) method [30] in **Table (4)**. The AMS construction time refers to the time spent on the construction of the multiscale restriction and prolongation operators and the global coarse system. The BILU construction time denotes the CPU time on LU factorization of local blocks and global incomplete LU factorization. The total time for TAMS is the sum of the solver time, AMS construction time, and BILU construction time. The time spent on initialization of the wirebasket system and the blocked system for BILU is not included.

Table (4) shows that TAMS is more efficient than AMG in most test cases with an exception of the anisotropic log-normal permeability case. Moreover, the local LU factorization of BILU and construction of the prolongation operator (the most expensive part of AMS) are perfectly parallelizable. Therefore, TAMS is expected to be quite efficient for scalable, parallel computation.

The TAMS algorithm s epxected be quite effective in time dependent, multiphase flow problems with coupled flow and transport. Moreover, adaptive computation can be introduced to further improve the efficiency. For example, in the construction of the prolongation operator, only local regions that experience large mobility changes need to be recomputed. Furthermore, since the TAMS method has the capability to ensure local mass conservation, a relatively large convergence tolerance for the pressure solution can be taken, while still maintaining accuracy in solving the multiphase transport equations.

		log-normal		SPE 10	
		Isotropic	Anisotropic	Isotropic	Anisotropic
AMG	solver time	43.6	17.0	46.4	56.1
TAMS	AMS construction time	9.3	9.8	9.1	9.0
	BILU construction time	2.6	2.6	3.2	3.3
	solver CPU time	8.1	19.2	14.5	26.5
	total time	20.0	31.6	26.8	38.8

Table 4—Comparison of CPU time for TAMS and AMG in the efficiency test cases.

Scalability Analysis

Scalability is an important measurement of the performance of an algorithm for large-scale problems. For TAMS, there two important aspects. The first is how the iterations and CPU time depend on the problem sizes. The second is, for a given fine-scale problem, how does the performance depend on the coarsening ratio.

In the first set, the coarsest grid is $32 \times 32 \times 32$ and the logarithm of permeability is generated by sequential Gaussian simulation [11] from a Gaussian distribution with a mean and variance of 4. Then, the permeability is refined by a factor of two in each direction. In total, three different grids are generated, namely, $32 \times 32 \times 32$, $64 \times 64 \times 64$, and $128 \times 128 \times 128$ respectively. The permeability fields are shown in **Fig.(27)**. The same boundary conditions are used as in the efficiency study. We fix the coarsening ratio at $8 \times 8 \times 8$ for the three grids. The GMRES iteration numbers and CPU times are shown in **Table (5)**. The Galerkin restriction operator and BILU with block size $4 \times 4 \times 4$ are used for TAMS. 5 indicates that the convergence is practically insensitive to problem size. **Fig.(28)** shows the CPU time as a function of problem size. The CPU times and problem sizes are all normalized by the values for the coarsest grid. The increases in CPU time is almost linear with problem size. The slope for the solver time is slightly smaller than unity. This result shows that TAMS is perfectly scalable with the problem size.

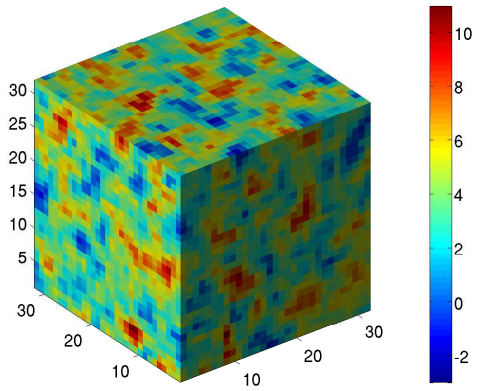


Fig. 27—The natural logarithm of heterogeneous permeability for scalability study.

In the second test case, we take the above grid with $128 \times 128 \times 128$ fine cells, and consider three coarsening ratios: $4 \times 4 \times 4$, $8 \times 8 \times 8$ and $16 \times 16 \times 16$. The GMRES iteration numbers and CPU times are shown in **Table (6)**. It is not surprising that increasing the coarsening ratio reduces the convergence rate. For larger coarsening ratios, the number of coarse cells is also smaller and the reduced boundary conditions on the dual coarse-grid have a larger global influence, which deteriorates the accuracy in the AMS step. Moreover, each dual block becomes larger, and the construction of AMS is thus more expensive. On the other hand, we do not want to use too small

problem size	$32 \times 32 \times 32$	$64 \times 64 \times 64$	$128 \times 128 \times 128$
GMRES iterations	12	10	8
solver time	0.43	3.03	24.5
AMS construction time	0.14	1.24	11.27
BILU construction time	0.30	2.66	21.62

Table 5—Iteration numbers and CPU time for different sizes of problems

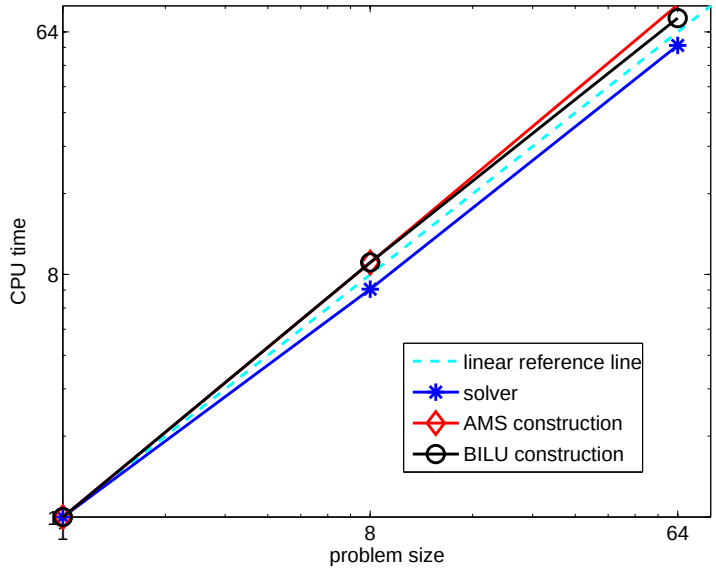


Fig. 28—The increase of CPU time with problem sizes. The CPU time and problems size are normalized by those of the $32\times32\times32$ problem

coarsening ratio, since the coarse system becomes larger and each iteration gets more expensive. An optimal choice of the coarsening ratio is problem dependent and may require experience, or a trial and error, to obtain. From our numerical experience, it is usually a good starting point to choose the coarsening ratio in each coordinate direction to be approximately the square root of the fine grid dimension in that direction.

coarsening ratio	$4 \times 4 \times 4$	$8 \times 8 \times 8$	$16 \times 16 \times 16$
GMRES iterations	6	8	16
solver time	26.9	24.5	44.4
AMS construction time	8.5	11.3	34
BILU construction time	21.5	21.6	21.5

Table 6—Iteration numbers and CPU time for different coarsening ratios.

Conclusions

An efficient two-stage algebraic multiscale solver (TAMS) algorithm is proposed in this paper. The TAMS method consists of two stages: one global and one local. In the global stage, a multiscale solution is obtained purely algebraically from the fine-scale matrix. An algebraic prolongation operator is obtained from a wirebasket ordered reduced system from the fine-scale matrix. Two restriction operators are defined: one is based on (conservative) finite-volume discretization and the other is based on the Galerkin scheme. In the second stage, a local preconditioner, such as BILU(0), or Additive Schwarz, is used.

Spectral analysis shows that the TAMS scheme approximates the full spectra of the original fine-scale matrix quite well and convergence to the fine-scale solution is guaranteed. Moreover, the TAMS spectra tend to cluster, and this is advantageous for Krylov subspace methods to achieve fast convergence. The TAMS method can also preserve the important property of local mass conservation of the multiscale finite volume method; this is achieved if a multiscale solution step with the finite-volume type of restriction operator is applied as the last 'clean up' step. This allows for the TAMS algorithm to be used in an adaptive framework for coupled multiphase

flow and transport.

Challenging large-scale problems with complex heterogeneous structure and high anisotropy ratio were investigated. The Galerkin type of restriction operator and the BILU local preconditioner are the most robust and efficient choices for the TAMS algorithm.

The performance of TAMS is comparable to the state-of-the-art algebraic multigrid (AMG) preconditioner. TAMS is amenable to massive parallelism both in the construction of the multiscale operators and the factorization of the BILU blocks. Moreover, the CPU time of the TAMS scheme is almost linear with problem size, which is very important for large-scale problems. The efficiency of TAMS can be further improved in time dependent problems with coupled flow and transport by adaptive computation of the prolongation operator and relaxed convergence tolerance for pressure as local mass conservation can always be achieved.

Acknowledgement

The authors gratefully acknowledge financial support from DOE grant DE-FG02-06-ER25727 and the members of the affiliate program of the Stanford University Petroleum Research Institute for Reservoir Simulation (SUPRI-B).

References

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

[2] J. E. Aarnes and T. Y. Hou. Multiscale domain decomposition methods for elliptic problems with high aspect ratios. *Acta Mathematicae Applicatae Sinica*, 18(1):63–76, 2002.

[3] I. Aavatsmark. An introduction to multipoint flux approximations for quadrilateral grids. *Computational Geoscience*, 6:405–432, 2002.

[4] T. Arbogast. Implementation of a locally conservative numerical subgrid upscaling scheme for two-phase darcy flow. *Computational Geosciences*, 6:453–481, 2002.

[5] T. Arbogast and S. L. Bryant. A two-scale numerical subgrid technique for waterflood simulations. *SPE Journal*, 7:446–457, 2002.

[6] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.0.0, Argonne National Laboratory, 2008.

[7] X. C. Cai, W. D. Gropp, and D. E. Keyes. A comparison of some domain decomposition algorithms for nonsymmetric elliptic problems. In *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 224–235, 1992.

[8] Y. Chen, L. J. Durlofsky, M. Gerritsen, and X. H. Wen. A coupled local-global upscaling approach for simulating flow in highly heterogeneous formations. *Advances in Water Resources*, 26:1041–1060, 2003.

[9] Z. Chen and T. Hou. A mixed finite element method for elliptic problems with rapidly oscillating coefficients. *Mathematic Computation*, 72:541–576, 2003.

[10] M. A. Christie and M. J. Blunt. Tenth SPE comparative solution project: A comparision of upscaling techniques. *SPE Reservoir Eval. & Eng.*, 4:308–317, 2001.

[11] C. V. Deutsch and A. G. Journel. *GSLIB: Geostatistical Software Library and User's Guide*. Oxford University Press, New York, 1998.

[12] L. J. Durlofsky. Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media. *Water Resources Research*, 27:699–708, 1991.

[13] I. G. Graham, P. O. Lechner, and R. Scheichl. Domain decomposition for multiscale pdes. *Numerische Mathmatik*, 106:589–626, 2006.

[14] H. Hajibeygi, G. Bonfigli, M. A. Hesse, and P. Jenny. Iterative multiscale finite-volume method. *Journal of Computational Physics*, 227:8604–8621, 2008.

[15] M. A. Hesse, B. T. Mallison, and H. A. Tchelepi. Compact multiscale finite volume method for heterogeneous anisotropic elliptic equations. *SIAM Multiscale Model. Simul.*, 7:934–962, 2009.

[16] T. Hou and X. H. Wu. A multiscale finite element method for elliptic problems in composite matrials and porous meida. *Journal of Computational Physics*, 134:169–189, 1997.

[17] P. Jenny, S. H. Lee, and H. A. Tchelepi. Multiscale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, 187:47–67, 2003.

[18] P. Jenny, S. H. Lee, and H. A. Tchelepi. Adaptive multiscale finite-volume method for multiphase flow and transportaion in porous media. *Mulitscale Modeling and Simulation*, 3:50–64, 2004.

[19] P. Jenny, S. H. Lee, and H. A. Tchelepi. Adaptive fully implicit multiscale finite-volume method for multiphase flow and transport in heterogeneous porous media. *Journal of Computational Physics*, 217:627–641, 2006.

[20] V. Kippe, J. E. Aarnes, and K. A. Lie. A comparison of multiscale methods for elliptic problems in porous media flow. *Computational Geoscience*, 12(3):377–398, 2008.

[21] S. H. Lee, C. Wolfsteiner, and H. A. Tchelepi. A multiscale finite-volume method for multiphase flow in porous media: Black oil formulation of compressible, three phase flow with gravity. *Computational Geosciences*, 12:351–366, 2008.

[22] S. H. Lee, H. Zhou, and H. A. Tchelepi. Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogeneous formations. *Journal of Computational Physics*, accepted, 2009.

[23] I. Lunati and P. Jenny. Multi-scale finite-volume method for highly heterogeneous porous media with shale layers. In *Proceedings of the 9th European Conference on the Mathematics of Oil Recovery (ECMOR)*, Cannes, Frace, 2004.

[24] I. Lunati and P. Jenny. Multi-scale finite-volume method for compressible multi-phase flow in porous media. *Journal of Computational Physics*, 216:616–636, 2006.

[25] I. Lunati and P. Jenny. Multi-scale finite-volume method for multi-phase flow with gravity. *Computational Geosciences*, 12: 337–350, 2008.

[26] J. M. Nordbottern and P. E. Bjørstad. On the relationship between the multiscale finite-volume method and domain decomposition preconditioners. *Computational Geosciences*, 12:367–376, 2008.

[27] D. W. Peaceman. *Fundamentals of Numerical Reservoir Simulation*. Elsevier Scientific Pub., 1977.

[28] Y. Saad. *Iterative methods for sparse linear system*. PWS Publishing, 1996.

[29] B. F. Smith, P. E. Bjørstand, and W. D. Gropp. *Domain Decomposition*. Cambridge University Press, 1996.

[30] K. Stüeben. *Multigrid*, chapter Algebraic Multigrid (AMG): An Introduction with Applications. Academic Press, 2000.

[31] H. A. Tchelepi, P. Jenny, S. H. Lee, and C. Wolfsteiner. Adaptive multiscale finite volume framework for reservoir simulation. *SPE Journal*, 12:188–195, 2007.

[32] A. Toselli and O. Wildlund. *Domain Decomposition Methods – Algorithms and Theory*. Springer, 2005.

[33] J. R. Wallis and H. A. Tchelepi. Apparatus, method and system for improved reservoir simulation using an algebraic cascading class linear solver. Technical report, U.S. Patent NO. 7684967, 2010.

[34] H. Zhou. Operator based multiscale method for compressible flow. Master's thesis, Stanford University, 2006.

[35] H. Zhou and H. A. Tchelepi. Operator based multiscale method for compressible flow. *SPE Journal*, 13:267–273, 2008.