

# Mixed Multiscale Methods for Compressible Flow

S. Krogstad (SINTEF)

K.-A. Lie\* (SINTEF)

B. Skaflestad (SINTEF)

September 13, 2012

## Abstract

Multiscale methods are a robust and accurate alternative to traditional upscaling methods. Multiscale methods solve local problems to numerically construct a set of basis functions that later can be used to compute global solutions that describe the flow on both the coarse computational scale and the underlying fine parameter scale. This way, one is able to account for both effective coarse-scale properties and sub-scale variations. The methods are particularly efficient when the flow field must be updated repeatedly. Because temporal changes in the flow equations are moderate compared to the spatial variability, it is seldom necessary to recompute basis functions each time the global flow field is recomputed. Herein, we discuss and compare two ways of extending a multiscale mixed method that was originally developed for incompressible flow to compressible flow. The first approach is based upon a mixed residual formulation with a fine-scale domain-decomposition corrector. The second approach is to associate more than one basis function for each coarse face and coarse cell and use bootstrapping to dynamically build a basis function dictionary that spans the evolving flow patterns. We present and discuss several numerical examples, from simplified 1D cases to 3D cases with realistic reservoir geometries.

## Introduction

For the oil industry to succeed in increasing oil recovery there is a growing trend for model-based decisions. New and exciting developments are seen in a variety of areas such as real-time reservoir management, uncertainty quantification, integrated operations, closed-loop management, and production optimization. Common to all these fields of endeavor is the requirement for fast flow simulation in which the simulation model is tightly coupled to the geology and to dynamic data sources. However, there is a significant, and increasing, gap between the level of detail seen in geological models and the capabilities of contemporary reservoir simulators. Instead of focusing on understanding the physical characteristics of reservoirs and the economic consequences of different developments, a lot of valuable human resources is diverted to upscaling (and downscaling) and its negative consequences for the representation of heterogeneities and fluid flow. Not only is upscaling costly, but in the process much of the information inherent in high-resolution geological models is lost when upscaled to a coarser simulation model in which local flow structures is only preserved in an averaged sense. Enabling the oil industry to make a step-change in its work processes therefore calls for a radical speedup of flow simulation and for simulators that are equipped to utilize both static data and the vast amount of dynamic data that becomes available.

There are several technological developments that can contribute to a radical speedup of flow simulation: advances in hardware, parallel algorithms, improved (non)linear solvers, and alternative formulations (streamlines, operator splitting), to name a few. Another important contribution may come from multiscale methods, as will be discussed herein. Generally speaking, multiscale methods are numerical methods and strategies that aim to describe the coupling of different physical phenomena on coarse grids while properly accounting for the influence of unresolved fine-scale structures in the porous media. However, unlike traditional upscaling techniques, multiscale methods often provide a mechanism to recover approximate fine-scale solutions.

The purpose of this paper is to develop a multiscale method that can simulate compressible flow in highly heterogeneous media described by stratigraphic and unstructured grids. Multiscale methods have previously been developed for compressible flow based upon a finite-volume formulation [12] that requires a primal and a dual grid and uses an iterative scheme [10, 11] to reduce the fine-scale residual. Constructing a dual grid explicitly is possible for realistic reservoirs but may be both costly and difficult for complex grid models [19]. Herein, we will therefore base our development on a multiscale mixed finite-element formulation [6, 8] that is more flexible to the geometry and topology of the grid and hence readily applicable to realistic geo-cellular models [4, 5]. To this end, we will present and compare two different approaches: The first method, iMsMFE for short, starts with a standard formulation of elliptic basis functions and introduces a set of localized residual functions and extra nonlinear iterations to resolve non-elliptic effects and drive the fine-scale residual towards zero. How to resolve the elliptic part of a compressible black-oil model was discussed in [15], and an early example of using the iMsMFE method was presented in [16]. The second method, bMsMFE, is based upon a new multiscale formulation that associates more than one basis function with each coarse block and coarse interface and uses a bootstrapping approach to dynamically build a basis function dictionary that adapts to the evolving parabolic solution. The bMsMFE method builds on ideas used previously by Krogstad [14] for model reduction. As part of our discussion, we also discuss a third method (cMsMFE) that is only applicable to genuinely compressible systems.

The software prototypes and all numerical examples presented in the following were developed using the Matlab Reservoir Simulation Toolbox [16], which is freely available as open source.

## Mathematical model and fine-scale discretization

To keep the discussion as simple as possible, we will present the various concepts using two different models. For the basic concepts of the multiscale method, we consider incompressible, immiscible flow in the absence of gravity and capillary forces:

$$\vec{v} = -\lambda \mathbf{K} \nabla p, \quad \nabla \cdot \vec{v} = q. \quad (1)$$

Here,  $\vec{v}$  denotes Darcy velocity,  $\lambda$  the relative mobility,  $\mathbf{K}$  the absolute permeability,  $p$  the fluid pressure, and  $q$  source terms. Likewise, to present how compressibility can be handled, we will consider a model written on the compact form:

$$\begin{aligned} \vec{v} &= -\lambda \mathbf{K} (\nabla p - \sum_j \rho_j f_j \vec{g}) \\ \nabla \cdot \vec{v} &= q - c_t \frac{\partial p}{\partial t} + \left( \sum_j c_j f_j \vec{v} + \alpha(p) \mathbf{K} \vec{g} \right) \cdot \nabla p \end{aligned} \quad (2)$$

where  $\rho_j$  denotes phase densities,  $f_j$  fractional flow functions,  $\vec{g}$  the gravity vector,  $c_j$  and  $c_t$  the phase and total compressibilities, and  $\alpha(p)$  a known function of pressure-dependent parameters.

The computational domain  $\Omega$  is partitioned into a set  $\{\Omega_i\}$  of  $n_c$  non-overlapping polyhedral cells so that cell  $i$  has  $n_i$  planar faces that match the faces of the cell's neighbors. Altogether, the grid has  $n_f$  faces, out of which  $n_f^i$  are interior to the grid. For the multiscale methods, we will form a second and coarser grid with grid blocks  $B_\ell$  that consist of a connected set of cells  $\Omega_i$  from the fine grid. The grid will consist of  $N_b$  blocks having a total of  $N_f$  faces, out of which  $N_f^i$  are interior to the grid. To discretize (1) on a general polyhedral cell, we will use methods written on the form

$$\mathbf{M} \mathbf{v}_i = p_i \mathbf{e}_i - \pi_i. \quad (3)$$

Here,  $\mathbf{v}_i$  denotes the vector of outward fluxes of the faces of  $\Omega_i$ ,  $p_i$  the pressure at the cell center,  $\mathbf{e}_i$  is a vector of ones,  $\pi_i$  the vector of face pressures, and  $\mathbf{M}$  is the inverse of the corresponding transmissibility matrix  $\mathbf{T}_i$ . Using this notation, the discretized incompressible model (1) can be either written on mixed or hybrid-mixed form, i.e.,

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ -\mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{q} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{B} & \mathbf{C} & \mathbf{D} \\ \mathbf{C}^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{D}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ -\mathbf{p} \\ \pi \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{q} \\ \mathbf{0} \end{bmatrix}, \quad (4)$$

where  $\mathbf{v}$  and  $\pi$  are  $n_f^i \times 1$  vectors containing the unknown face fluxes and face pressures and  $\mathbf{p}$  is a  $n_c \times 1$  vector with the unknown cell pressures. Equation (4) can be derived by augmenting (3) with flux continuity across cell faces for the mixed form and flux and pressure continuity for the hybrid form. Both forms may be used interchangeably in the following; the mixed-hybrid form has the advantage that it can be reduced to a symmetric positive-definite system using a block-wise Gaussian elimination. In a similar manner, we can linearize the compressible model (2) and introduce our standard discretization (3) to derive the following mixed discrete system

$$\begin{bmatrix} \mathbf{B}(\mathbf{s}^n) & \mathbf{C} \\ \mathbf{C}^\top & \mathbf{P}(\mathbf{s}^n, \mathbf{p}_{v+1}^{n+1}) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{v+1}^{n+1} \\ -\mathbf{p}_{v+1}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{s}^n, \mathbf{p}_v^{n+1}) \\ \mathbf{g}(\mathbf{s}^n, \mathbf{p}^n, \mathbf{p}_v^{n+1}) \end{bmatrix}, \quad (5)$$

where  $n$  denotes time steps and  $v$  iteration steps.

## Multiscale mixed finite elements – key algorithmic components

To derive the MsMFE methods for (1) or (2) we start by decomposing the solution to (4) or (5) as follows

$$\mathbf{v} = \Psi \mathbf{v}_c + \tilde{\mathbf{v}}, \quad \mathbf{p} = \Phi \mathbf{p}_c + \tilde{\mathbf{p}}, \quad \pi = \Pi \pi_c + \tilde{\pi}. \quad (6)$$

Here,  $\mathbf{u}_c$  denotes the vector of outward fluxes over the coarse-block interfaces,  $\mathbf{p}_c$  denotes the vector of coarse-block pressures, and  $\pi_c$  denotes the vector of coarse-block face pressures. Likewise,  $\tilde{\mathbf{u}}, \tilde{\mathbf{p}}, \tilde{\pi}$  are reminder terms having variations on the fine grid. The matrices  $\Psi, \Phi$ , and  $\Pi$  represent the fine-scale reconstruction operators for  $\vec{v}, p$ , and  $\pi$ . Altogether, we have therefore defined a reconstruction operator  $\mathbf{R} = \text{diag}(\Psi, \Phi, \Pi)$  that brings us from the degrees-of-freedom  $\mathbf{x}_c = [\mathbf{u}_c, -\mathbf{p}_c, \pi_c]$  on the coarse-scale to those on the fine scale  $\mathbf{x} = [\mathbf{u}, -\mathbf{p}, \pi]$ .

Each column in  $\Psi$  corresponds to a multiscale basis function for the flux associated with a unique coarse-grid face and is represented as a  $n_f \times 1$  vector of fine-scale fluxes. There are several ways to construct these basis functions, as we will see next. For compressible flow, we also need to define fine-scale variations for the pressure bases so that each column of  $\Phi$  corresponds to a basis function associated with a unique cell and each column of  $\Pi$  corresponds to a basis function defined over a coarse face. For incompressible flow, the pressure is immaterial and is seldom used except in well models. This means that  $\Phi$  can be replaced by a simple prolongation operator  $\mathbf{I}$  that maps a constant value from each coarse block and onto the cells of the block. Likewise,  $\Pi$  is replaced by a prolongation operator  $\mathbf{J}$  that maps a constant value from each coarse face and onto the cell faces that make up the coarse face.

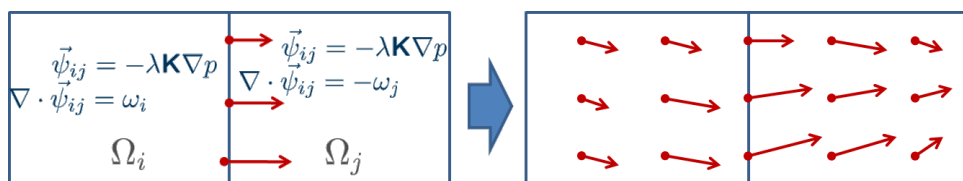
**One-block approach: local and global** In the simplest setup, there is only one flux basis function per coarse-block interface and this basis function can be written as a sum of two parts  $\vec{\psi}_{ij} = \vec{\psi}_{ij} - \vec{\psi}_{ji}$ . Each component  $\vec{\psi}_{ij}$  is localized to a single block  $B_i$ , in which it solves

$$\vec{\psi}_{ij} = -\lambda \mathbf{K} \nabla \phi_{ij}, \quad \nabla \cdot \vec{\psi}_{ij} = w_i(\mathbf{x}), \quad \mathbf{x} \in B_i. \quad (7)$$

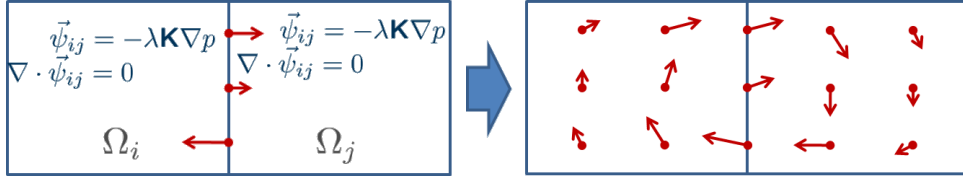
Here,  $w_i$  is a positive weight function used to drive flow inside the block. (In practice, the two parts can be computed by solving one problem over the two block  $B_i$  and  $B_j$  as shown in Figure 1). For now, we can assume that  $w_i(\mathbf{x}) \equiv |B_i|^{-1}$ ; we will come back to other choices later. To close the system, we specify a set of Neumann boundary conditions

$$(\vec{\psi}_{ij} \cdot \vec{n}_i)(\mathbf{x}) = \frac{v_{ij}(\mathbf{x})}{\int_{\Gamma_{ij}} v_{ij}(s) ds}, \quad \mathbf{x} \in \Gamma_{ij}, \quad (\vec{\psi}_{ij} \cdot \vec{n}_i)(\mathbf{x}) = 0, \quad \mathbf{x} \in \partial B_i \setminus \Gamma_{ij}. \quad (8)$$

Here,  $\vec{n}_i$  denotes the normal to the interface  $\partial B_i$  and  $\Gamma_{ij} = \partial B_i \cap \partial B_j$  the interface between blocks  $B_i$  and  $B_j$ . The boundary condition  $v_{ij}$  specifies the flow out of the block and determines the basis function uniquely for a given function  $w$ . It is therefore essential that  $v_{ij}$  reflects the dominating features that impact the local flow behavior, e.g., heterogeneous structures that penetrate the interface and lead to uneven distribution of flux. Two alternative approaches can be used to determine  $v_{ij}$  [1, 2]. In the simplest approach, the flow over the interface is set proportional to the relative mobility on the interface,  $v_{ij} = \vec{n}_{ij} \cdot \lambda \mathbf{K} \vec{n}_{ij}$ , thereby providing a set of **local boundary conditions**. Alternatively, if we know the global solution  $\vec{v}^0$ , or a good approximation thereof, we can use this solution as a set of **global boundary conditions**,  $v_{ij} = \vec{v}^0 \cdot \vec{n}_{ij}$ , that in addition to reflecting local heterogeneity also incorporates far-field effects. Using global boundary conditions, one is generally able to exactly reproduce the global field  $\vec{v}^0$ . As an alternative to global conditions, one can use a so-called local-global approach in which a bootstrapped sequence of global solutions on upscaled models are used estimate good boundary conditions, see e.g., Alpak et al. [5].



**Figure 1** Construction of a basis function using the one-block approach in which a Neumann boundary condition (red color) is prescribed on each coarse-block interface. No-flow conditions are prescribed on all other boundaries.



**Figure 2** Construction of a divergence-free basis function, for which a Neumann boundary condition (red color) with zero net flux is prescribed on each coarse-block interface. No-flow conditions are prescribed on all other boundaries.

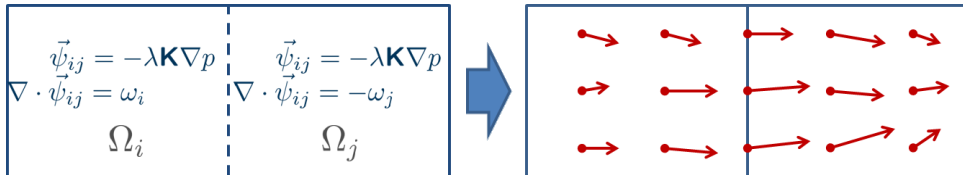
**Divergence-free basis functions.** In some cases, we may wish to associate more than one flux basis function to each coarse interface. To avoid instabilities in the resulting coarse-scale system it is a good idea to construct the extra basis functions so that they are divergence free on the coarse scale [14]; that is, so that the fine-grid fluxes over  $\Gamma_{ij}$  sum to zero. To this end, we set  $\tilde{\psi}_{ij} = \tilde{\psi}_{ij} - \tilde{\psi}_{ji}$ , where each component  $\tilde{\psi}_{ij}$  solves a local *homogeneous* equation

$$\tilde{\psi}_{ij} = -\lambda \mathbf{K} \nabla \phi_{ij}, \quad \nabla \cdot \tilde{\psi}_{ij} = 0, \quad \mathbf{x} \in B_i, \quad (9)$$

with boundary conditions

$$\tilde{\psi}_{ij} \cdot \vec{n}_i = \frac{v_{ij} - \int_{\Gamma_{ij}} v_{ij} ds}{\int_{\Gamma_{ij}} v_{ij} ds} \quad \text{on } \Gamma_{ij}, \quad \tilde{\psi}_{ij} \cdot \vec{n}_i = 0 \quad \text{on } \partial B_i \setminus \Gamma_{ij}. \quad (10)$$

The only difference from (7) and (8) is that because there is no source term, the net flux across  $\Gamma_{ij}$  will be zero. This means that the resulting basis function will be a local rotational field, see Figure 2. The basis function cannot represent net flow between  $B_i$  and  $B_j$  and must hence be complemented with other basis functions that have this ability.



**Figure 3** Construction of a basis function using the two-block approach, for which no condition is prescribed on the coarse-block interface. No-flow conditions are prescribed on all other boundaries.

**Two-block approach.** The global method discussed above is generally quite accurate, but requires the knowledge of a fine-grid solution. The local method, on the other hand, may give inaccurate approximations to the velocity, see e.g., [2]. As an intermediate solution, Aarnes et al. [3] suggested to use a two-block approach in which no condition is placed on the interface to which the basis function is associated. Consider two neighboring blocks  $B_i$  and  $B_j$ , and let  $B_{ij}$  be a singly-connected subset of  $\Omega$  that contains  $B_i$  and  $B_j$ . A multiscale basis function associated with the interface  $\Gamma_{ij} = \partial B_i \cap \partial B_j$  can be computed by solving

$$\tilde{\psi}_{ij} = -\mathbf{K} \nabla \phi_{ij}, \quad \nabla \cdot \tilde{\psi}_{ij} = w_i(\mathbf{x}) \begin{cases} w_i(\mathbf{x}), & \text{if } \mathbf{x} \in B_i, \\ -w_j(\mathbf{x}), & \text{if } \mathbf{x} \in B_j, \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

in  $B_{ij}$  with  $\tilde{\psi}_{ij} \cdot \vec{n} = 0$  on  $\partial B_{ij}$ , see Figure 3. If  $B_{ij} \neq B_i \cup B_j$ , we say that the basis function is computed using **overlap** or **oversampling** to lessen the impact of the artificial no-flow condition on the boundary.

**Computing basis functions.** All basis functions are independent of each other, regardless of whether they are determined by (7) or (11), and can therefore be computed in parallel. Alternatively, the local systems can be collected into a matrix equation on the form,

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Psi \\ -\Phi \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{w} \end{bmatrix}, \quad (12)$$

where  $\Psi$  is a  $n_f^i \times N_f^i$  matrix containing the fine-scale fluxes in the multiscale basis functions,  $\Phi$  is a  $n_c \times N_b$  matrix with the corresponding fine-grid pressure values, and  $\mathbf{w}$  is a  $n_c \times N_b$  matrix with the weight functions.

**Coarse-scale system.** Having defined the multiscale decomposition, the next thing we need is a compression operator that we can pre-multiply the global fine-scale system (6) with to reduce it to a global coarse-scale system. We notice that the transpose of the prolongation operator  $\mathbf{I}$  corresponds to the sum over all cells. Hence, we choose  $\mathbf{R}^\top$  as our compression operator and obtain

$$\begin{bmatrix} \Psi^\top \mathbf{B} \Psi & \mathbf{I}^\top \mathbf{C} \mathbf{I} \\ \mathbf{I}^\top \mathbf{C}^\top \Psi & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_c \\ -\mathbf{p}_c \end{bmatrix} = \begin{bmatrix} -\Psi^\top \mathbf{B} \tilde{\mathbf{v}} + \Psi^\top \mathbf{C} \tilde{\mathbf{p}} \\ \mathbf{q}_c - \mathbf{I}^\top \mathbf{C}^\top \tilde{\mathbf{v}} \end{bmatrix}. \quad (13)$$

Here, we can make some further simplifications of the right-hand side. First, we observe that the second block equation in (12) gives us that  $\Psi^\top \mathbf{C} = \mathbf{w}^\top$ . Because the pressure is immaterial in an incompressible flow model, we can pick  $\tilde{\mathbf{p}}$  such that  $\mathbf{w}^\top \tilde{\mathbf{p}} = 0$ , which defines a unique splitting  $\mathbf{I} \mathbf{p}_c + \tilde{\mathbf{p}}$  and implies that the coarse-scale pressure is the  $w$ -weighted average of the true pressure,  $p_c^i = \int_{B_i} w p d\mathbf{x}$ . Neglecting the reminder term for the flux (and the face pressures), we obtain the following coarse-scale systems on mixed and mixed-hybrid form

$$\begin{bmatrix} \Psi^\top \mathbf{B} \Psi & \Psi^\top \mathbf{C} \mathbf{I} \\ \mathbf{I}^\top \mathbf{C}^\top \Psi & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_c \\ -\mathbf{p}_c \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{q}_c \end{bmatrix}, \quad \begin{bmatrix} \Psi^\top \mathbf{B} \Psi & \Psi^\top \mathbf{C} \mathbf{I} & \Psi^\top \mathbf{C} \mathbf{J} \\ \mathbf{I}^\top \mathbf{C}^\top \Psi & \mathbf{0} & \mathbf{0} \\ \mathbf{J}^\top \mathbf{D}^\top \Psi & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_c \\ -\mathbf{p}_c \\ \pi_c \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{q}_c \\ \mathbf{0} \end{bmatrix}. \quad (14)$$

**The weight function.** There are several ways to design the weight function so that it will drive flow inside each coarse block in a manner that mimics the physics of the underlying fine-scale equations. First, we observe that to produce a unit flow across the interface  $\Gamma_{ij}$ , the weight function should be chosen on the form  $w_i(\mathbf{x}) = \theta(\mathbf{x}) / \int_{B_i} \theta(\mathbf{x}) d\mathbf{x}$ . Secondly, a simple calculation shows the role of the weight function  $w_i(\mathbf{x})$  is to distribute  $\int_{B_i} \nabla \cdot \vec{v}$  in varying proportions to all cells within the block in an appropriate way so that the result mimics the fine-grid divergence. For incompressible flow,  $\text{div}(\vec{v})$  is nonzero only in grid blocks containing a nonzero source  $q$  and hence we should set  $\theta = q$  in these blocks to ensure mass-conservative velocity fields on the subgrid scale. Elsewhere, one can set  $\theta = 1$  or  $\theta = \text{trace}(\lambda \mathbf{K})$ . For compressible flow, the divergence of the velocity reads

$$\nabla \cdot \vec{v} = q - c_t \frac{\partial p}{\partial t} - \left( \sum_j c_j f_j \vec{v} + \alpha(p) \mathbf{K} \vec{g} \right) \cdot \nabla p. \quad (15)$$

Ideally,  $\theta$  should therefore be chosen proportional to the right-hand side of (15). In general, this will require updating  $\theta$  and recomputing the multiscale basis functions, which may or may not be feasible, depending on the number of updates compared with the total number of iterations and time steps required to capture the dynamics of the problem. Alternatively, one may set  $\theta = \phi$ , which can be motivated if the accumulation term dominates the right-hand side of (15). Because  $p$  is piecewise constant on the coarse grid, the accumulation term is locally proportional to  $c_t$ , which again is proportional to  $\phi$  for smooth saturations. Moreover, by choosing  $\theta = \phi$ , one should (to some extent) avoid forcing too much flow through low permeable barriers since regions with very low permeability tend to also have low porosity.

**Temporal dependence** For multiphase flow applications, the basis functions are generally time-dependent and coupled to the transport equation(s) through the relative mobility term  $\lambda$ , which is a function of the unknown and time-dependent saturations. However, this temporal dependence is typically quite weak and good multiscale solutions can be computed using infrequent updating of basis functions. Thinking of a Buckley–Leverett type displacement profile,  $\lambda(\mathbf{x}, t)$  will typically only vary modestly before and after the block is swept by a strong saturation (or component) front. When the front passes through a block, it may be necessary to recompute the corresponding basis functions to account for strong mobility variations in the interior of the block. Favorable displacements will typically contain strong fronts, and here the basis functions need to be updated more frequently than for unfavorable displacements, for which it is often sufficient to only compute the basis functions initially [13]. When using *global* boundary conditions, it is also necessary to scale the boundary conditions to account for variations in total mobility, i.e., use  $v_{ij} = (\lambda/\lambda^0) \bar{v}^0 \cdot \bar{n}_{ij}$ , where  $\lambda^0$  denotes the total mobility distribution associated with  $\bar{v}^0$ .

**Subscale pressure variation.** Subscale variation in pressure is essential to simulate models that include PVT behaviour, e.g., as in the (compressible) black-oil models. When using subscale variations in the pressure basis,  $\Phi$  and  $\Psi$  should scale similarly. To see this, we consider the first block equation in the definition of the basis functions (12), from which we have that

$$\mathbf{B}\Psi - \mathbf{C}\Phi = \mathbf{0} \implies \mathbf{B}\Psi \mathbf{v}_c - \mathbf{C}\Phi \mathbf{v}_c = \mathbf{0}.$$

Hence, the starting-point for the algebraic reduction should be the following fine-scale system

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Psi \mathbf{v}_c \\ -\mathbf{I} \mathbf{p}_c - \Lambda \Phi \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{q} \end{bmatrix},$$

where the diagonal matrix  $\Lambda = \text{diag}(\lambda_i^0/\lambda_i)$  accounts for variations in total mobility  $\lambda$ . For incompressible flow, on the other hand, pressure is immaterial and the (small) extra computational cost associated with assembling pressure basis functions can therefore be avoided in most applications.

**Residual corrections.** The multiscale method will generally not compute a solution with zero fine-grid residual. This may be caused by incorrect boundary conditions in both the one-block and two-block approaches or if more physical effects are included in the global (coarse-scale) system than those used to compute basis functions. To get a convergent method, we need to also account for variations that are not captured by the basis functions. We therefore formulate a system for the residual terms  $\tilde{\mathbf{v}}$  and  $\tilde{\mathbf{p}}$ ,

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{v}} \\ -\tilde{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} (\mathbf{C}\Lambda\Phi - \mathbf{B}\Psi) \mathbf{v}_c + \mathbf{C} \mathbf{I} \mathbf{p}_c \\ \mathbf{q} - \mathbf{C}^\top \Psi \mathbf{v}_c \end{bmatrix}. \quad (16)$$

This system can be solved directly, but in most cases we prefer to use a standard (non)overlapping domain-decomposition method. To this end, we introduce a new coarse grid  $\{\hat{B}_i\}$  defined such that  $B_i = \hat{B}_i$  if we use a non-overlapping method and  $B_i \subset \hat{B}_i$  otherwise. The residual equation (16) is then solved locally with no-flow boundary conditions on  $\hat{B}_i$  and zero right-hand side in  $\hat{B}_i \setminus B_i$  to generate an approximate residual

$$\mathbf{v} \approx \mathbf{v}^{\text{ms}} + \hat{\mathbf{v}}, \quad \hat{\mathbf{v}} = \sum_i \hat{\mathbf{v}}_i, \quad \mathbf{p} \approx \mathbf{p}^{\text{ms}} + \hat{\mathbf{p}}, \quad \hat{\mathbf{p}} = \sum_i \hat{\mathbf{p}}_i. \quad (17)$$

The convergence of such an domain-decomposition method can be quite slow, and to get an efficient overall method one typically needs the multiscale approximation to be close to the true solution. We will continue to discuss how to combine residual corrections with multiscale methods in the next section.

## An iterative multiscale mixed finite-element method

In [15], we presented what was our first attempt to make a mixed multiscale method for compressible black-oil models. The multiscale method was a direct extension of MsMFE as defined for incompressible flow, using a fixed set of basis functions that are not updated during the simulation. The basis functions are computed with  $\theta = \phi$  and include subscale pressure variations. The resulting method is quite simple and will in most cases give reasonable results for problems with weak compressibility and no phase changes. If the dynamics of the problem is dominated by the heterogeneity, the MsMFE method will provide a good initial guess and only a few extra iterations will be required to reconcile physical effects (compressibility). Unfortunately, the method does not work well for problems for which compressibility effects are dominant. Instead, we therefore propose an iterative predictor-corrector method, iMsMFE for short, that consists of the following four components:

- Elliptic basis functions, defined by either (7) or (11) and constructed with  $\theta = \text{trace}(\lambda \mathbf{K})$  for purely incompressible flow and  $\theta(\mathbf{x}) \propto \phi(\mathbf{x})$  for compressible flow.
- The following coarse-scale system for the multiscale predictor

$$\begin{bmatrix} \Psi^T \mathbf{B} \Psi & \Psi^T \mathbf{C} \mathbf{I} \\ \mathbf{I}^T (\mathbf{C}^T \Psi - \mathbf{P}_v \Lambda \Phi) & \mathbf{I}^T \mathbf{P}_v \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_c^{v+1} \\ -\mathbf{p}_c^{v+1} \end{bmatrix} = \begin{bmatrix} \Psi^T \mathbf{f}_v \\ \mathbf{I}^T \mathbf{g}_v \end{bmatrix}. \quad (18)$$

- The following system for the residuals

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{P} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{v}}^{v+1} \\ -\hat{\mathbf{p}}^{v+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_c - \Psi^T \mathbf{B} \Psi \mathbf{v}_c + \Psi^T \mathbf{C} \mathbf{I} \mathbf{p}_c \\ \mathbf{g}_c - \mathbf{I}^T (\mathbf{C}^T \Psi - \mathbf{P}_v \Lambda \Phi) \mathbf{v}_c + \mathbf{I}^T \mathbf{P}_v \mathbf{I} \mathbf{p}_c \end{bmatrix} \quad (19)$$

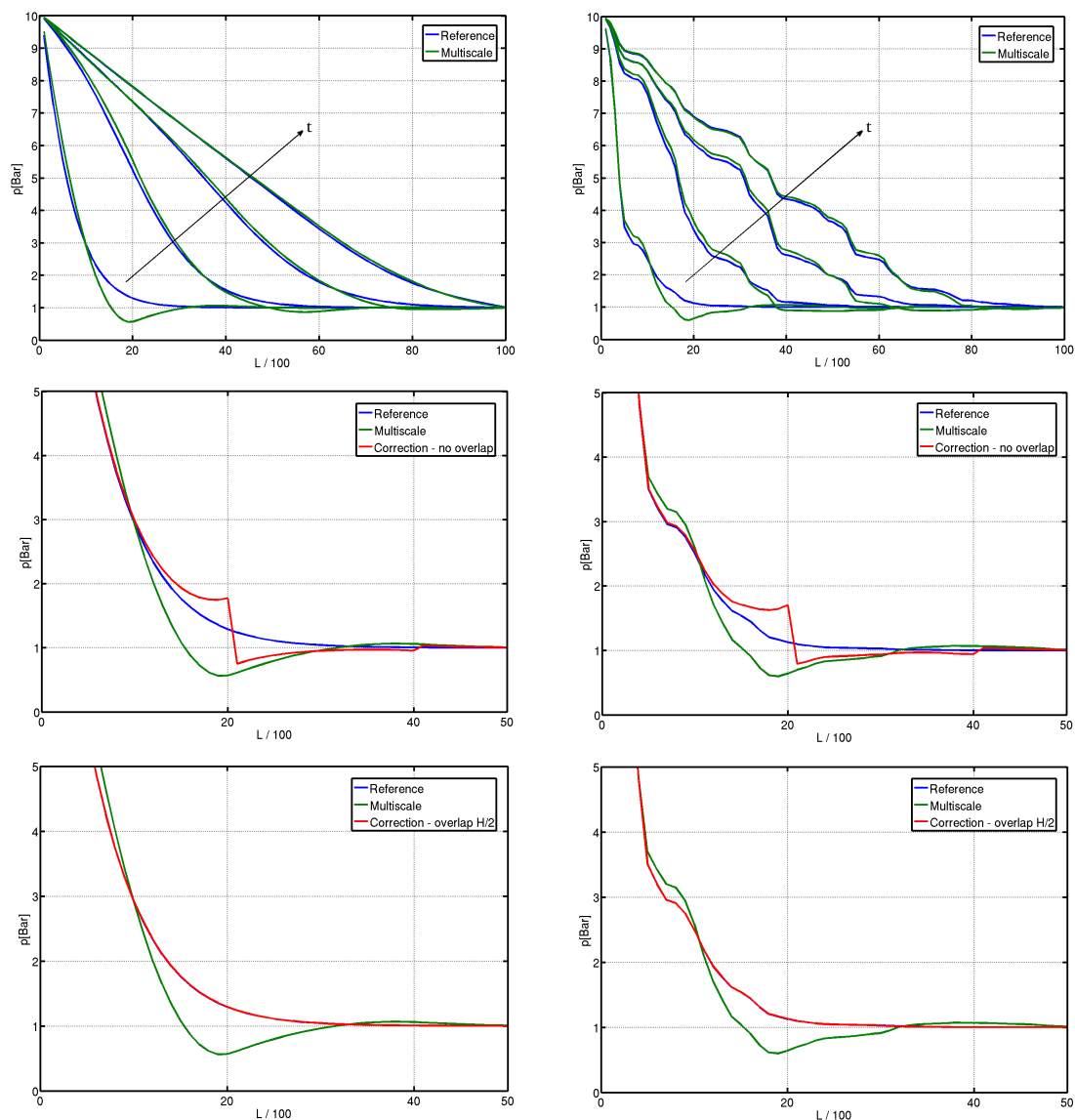
solved using an overlapping domain-decomposition method.

- Iterations over the multiscale and residual equations.

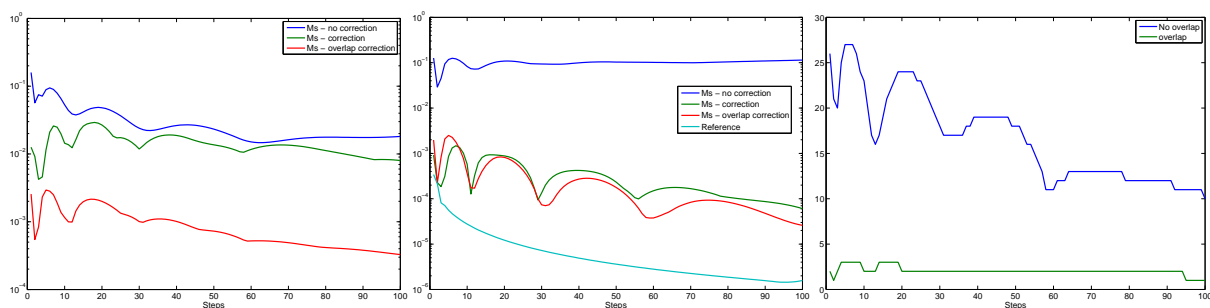
The choice of  $\theta = \phi$ , means that compressibility is only accounted for on the coarse scale (or that the elliptic basis functions mostly account for rock compressibility), which is 'correct' for block-constant pressure drop in slightly compressible flow. In our experience, this choice is reasonable for weakly compressible flows or when the aim of using the multiscale method is not to reproduce the solution of the fine-scale equation, but rather to account for subgrid heterogeneities influencing a flow resolved on the coarse grid. In the general case, however, compressibility effects can only be properly accounted for by the local residual functions that are used as *correction* functions to systematically drive the fine-grid residuals to zero.

To demonstrate the method, we consider a simple test case proposed by Lunati and Jenny [17] (for more results, see [20]), which greatly emphasizes the effects of compressibility and consequently presents a challenging case for our numerical method, because of the relatively low pressures involved, at least compared with realistic reservoir pressures. For 1D problems, the pressure pulse has a uniquely identifiable direction in which the corrections can sweep to reduce the coarse-scale error. In particular, this means that the fine-scale residual can be reduced to zero by a single Gauss-Seidel sweep (multiplicative Scharz domain decomposition). This is not possible in higher dimensions and to provide a more representative estimate of the efficiency of the method we will in the following example instead use an additive (Jacobi-type) approach, in which the residual functions are computed independently.

**Example 1** Consider a one-dimensional reservoir that is initially at atmospheric pressure ( $p = 1$  bar) and saturated with air. At the left boundary we inject an ideal tracer at a constant pressure of  $p = 10$  bar, while the pressure at the right boundary is kept fixed at  $p = 1$  bar. The domain is partitioned into one hundred cells of equal size and five equally-sized coarse grid blocks. The upper row in Figure 4 shows the resulting pressure signal after 1, 10, 50, and 100 pressure steps for a homogeneous and a lognormal



**Figure 4** Test of compressibility effects in the MsMFE and iMsMFE methods for an ideal gas. The upper row shows MsMFE solutions compared with fine-scale solutions for a homogeneous (left) and lognormal (right) reservoir. The next two rows show iMsMFE solutions computed with no overlap (middle row) and half a block-length overlap (lower row) in the residual corrections.



**Figure 5** Improved resolution obtained by adding correction functions to the MsMFE method. The left and middle plot show the reduction in maximum pressure discrepancy and average relative mass error after one pass with corrections as a function of time step. The right plot shows the number of outer iterations needed to reduce the relative pressure discrepancy below  $10^{-5}$ .

permeability field. The time increment in each pressure step is  $\tau/400$ , where  $\tau = \mu\phi L^2/(k\Delta p)$  is an estimate of the time required to reach steady state. Although the multiscale approximation to the pressure signal is not monotonous in the early stages of the process, in particular for the heterogeneous case, the solution quickly adjusts to provide reasonable agreement with the reference solution. Hence, there is a certain quality in the overall pressure signal when viewed as a single-phase problem. In a multiphase setting, however, the multiscale solution has inconsistencies between pressure and fluxes on the fine scale that unfortunately will induce relatively large mass errors in a standard fine-scale transport solver, which is a much more serious deficiency than the relative differences in cell pressures observed in Figure 4. Likewise, the coarse-scale fluxes are also inaccurate and MsMFE will therefore not work as a good multiphase upscaling method either.

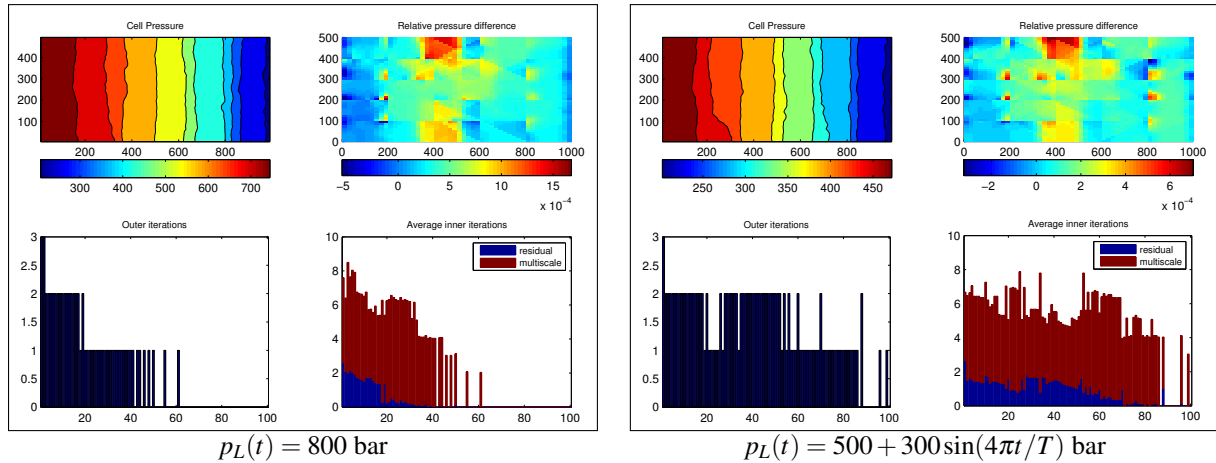
The two last rows in Figure 4 show a zoom of the first time step for the iterative multiscale solution before and after corrections. The correction functions clearly reduce inconsistencies between flux and pressure on the fine scale. Likewise, using overlap in the correction functions removes errors on the coarse scale. Figure 5 shows the maximum relative discrepancy in pressure and the average mass error as a function of time for correction functions computed with and without overlap equal one half coarse block. Introducing corrections improves the pressure discrepancy by one order of magnitude (for the early time steps) and the mass errors by two orders of magnitude. By using overlap, the discrepancy in pressure is reduced one order of magnitude further. Convergence towards the fine-scale solution can be ensured by feeding the corrections back into the multiscale system (18), to compute an updated multiscale solution, and so on. The right plot in Figure 5 shows the number of iterations required to reduce the relative pressure discrepancy below  $10^{-5}$ . With overlap, the desired resolution is obtained in 2–3 iterations, whereas if no overlap is used, the iterations converge rather slowly.

Updates to the correction functions will follow the propagation of the pressure signal, which means that the updates are localized. In higher dimensions, the updates will not necessarily be localized to a small part of the domain but one may still retain much of the same efficiency in many cases, as demonstrated in the next example.

**Example 2** We consider a  $1000 \times 500$  m reservoir described by a  $50 \times 50$  grid and a random permeability field spanning from 0.005 mD to 756 mD, with a mean value of 7.25 mD. The reservoir is initially filled with an ideal gas at 200 bar and produced from the right boundary at a constant pressure of 200 bar. We consider two different injection scenarios with gas being injected along the left boundary at a constant pressure of 800 or at oscillating pressure  $500 + 300 \sin(4\pi t/T)$  bar, where  $T$  denotes the total simulation time. Using a  $5 \times 5$  coarse grid, with 10 cells overlap in the residual functions, we simulate 100 time steps of length  $\tau/400$ , where  $\tau = \mu\phi L^2/(k\Delta p)$  is an estimate of the time required to reach steady state.

Figure 6 shows the pressure solutions, relative difference compared to the fine-scale solution at final time  $T = \tau/4$ , and the number of outer and inner iterations required to reduce the residual below  $10^{-3}$ . Iterations are required to resolve the propagating pressure pulses. However, once the solution approaches steady state after approximately 45 and 85 pressure steps, respectively, the solution can be computed by a single coarse-scale solve. The oscillatory boundary condition gives a longer period with pressure propagation and hence a higher number of overall iterations, whereas more iterations are needed in the first steps with constant injection pressure because the initial pressure difference is larger.

Table 1 reports a study of how the number iterations increases with a decreasing prescribed tolerance for four different degrees of overlap in the correction functions for the case with constant injection pressure. The multiscale predictor reduces the fine-scale residual below 0.1 for all steps except for the first and the number of iterations required to solve the global coarse-scale system increases linearly with decreasing tolerance and is almost insensitive to the degree of overlap. The computation of correction functions, on the other hand, is not computationally efficient as can be seen from the nonlinear increase in the



**Figure 6** Simulation by iMsMFE of an ideal gas injected at pressure  $p_L$  along the left boundary and produced at a constant pressure 200 bar along the right boundary. The lower plot shows the number of outer iterations per time step, the average number of additional nonlinear iterations required to solve the global coarse-scale system, and the average number of nonlinear iterations used per correction function.

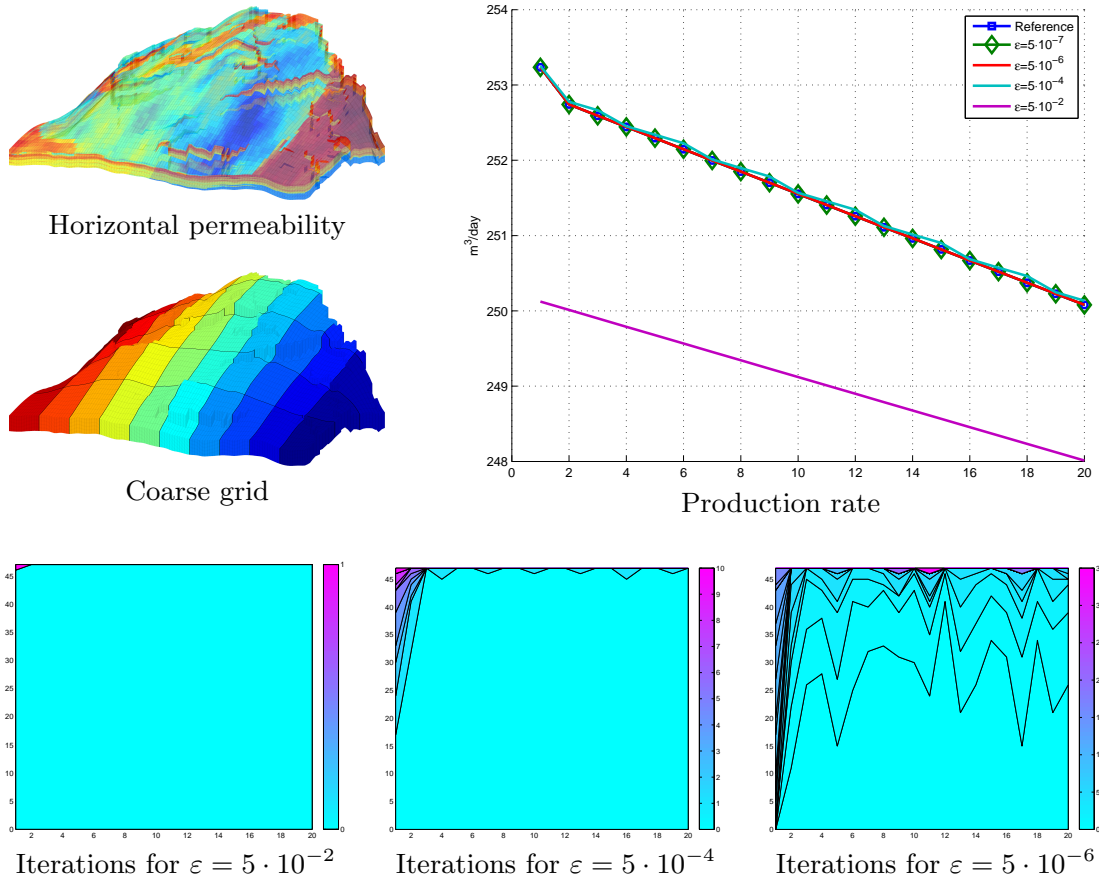
**Table 1** Average number of iterations per time step as a function of residual tolerance and degree of overlap in the correction functions.

	Overlap=10				Overlap=5				Overlap=3			Overlap=1	
	$10^{-3}$	$10^{-5}$	$10^{-7}$	$10^{-9}$	$10^{-3}$	$10^{-5}$	$10^{-7}$	$10^{-9}$	$10^{-3}$	$10^{-5}$	$10^{-7}$	$10^{-3}$	$10^{-5}$
Outer	0.68	1.89	3.52	5.92	0.81	3.30	8.68	17.90	0.81	4.13	13.35	2.85	32.62
Global	2.27	4.08	5.74	6.31	2.12	3.72	4.89	4.88	2.24	3.64	4.69	1.95	2.61
Correction	0.37	2.06	5.63	10.25	0.49	3.82	15.45	36.70	0.55	5.82	24.92	2.87	40.98

number of outer iterations and the number of inner iterations used in each local correction function. The nonlinear increase becomes more pronounced as the overlap is reduced, and with no overlap, the overall method fails to reduce the tolerance below 0.01.

In the next example, we apply the iMsMFE method to a fully realistic geological model to demonstrate that the method is capable of predicting well rates for the primary production of an ideal gas.

**Example 3 (Primary production)** We consider a synthetic, but highly realistic geological model from the SAIGUP project [18], see Figure 7. The reservoir is initially filled with an ideal gas at 200bar and produced from a single production well that is controlled by a bottom-hole pressure target of 150bar and completed in a single layer. The coarse model consists of 47 blocks formed as the result of a  $5 \times 10 \times 1$  partitioning of the logical cell indices. Using a correction function overlap of 12 cells we simulate 20 steps of production, each of 100 days, using different residual tolerances. All simulations predict the gradually diminishing production rate and, except for the least strict tolerance, generally agree with the result predicted by the reference solution. In particular, the predictions corresponding to the two finest tolerances are indistinguishable from that of the reference solution. The lower plots in Figure 7 shows the number of iterations for three of the simulations; the colored area below each graph shows the number of blocks that required zero, one, two, ... iterations to reduce the fine-scale residual below the prescribed tolerance. With an overly coarse tolerance of  $\epsilon = 5 \cdot 10^{-2}$ , one iteration is used in the well block and no iterations in the other blocks, which explains why this simulation fails to reproduce the correct production curve. With a more realistic tolerance of  $\epsilon = 5 \cdot 10^{-4}$ , up to ten iterations are used in the first time step, but from step three and on, zero iterations are used in almost all blocks. Altogether, the example shows that the accuracy can be significantly increased if one can afford the computational cost of a few extra correction functions.



**Figure 7** Simulation of primary production on a realistic geological model using the iMsMFE method.

The main problem with the current iterative method is that the multiscale predictor, which is set to only capture the 'elliptic' part of the dynamics, does not always do a good job and too much work is therefore left to the computationally inefficient, domain-decomposition corrector. Next, we will therefore present an approach for making basis functions that captures more of the parabolic nature of the problem.

### An MsMFE method for purely compressible flow

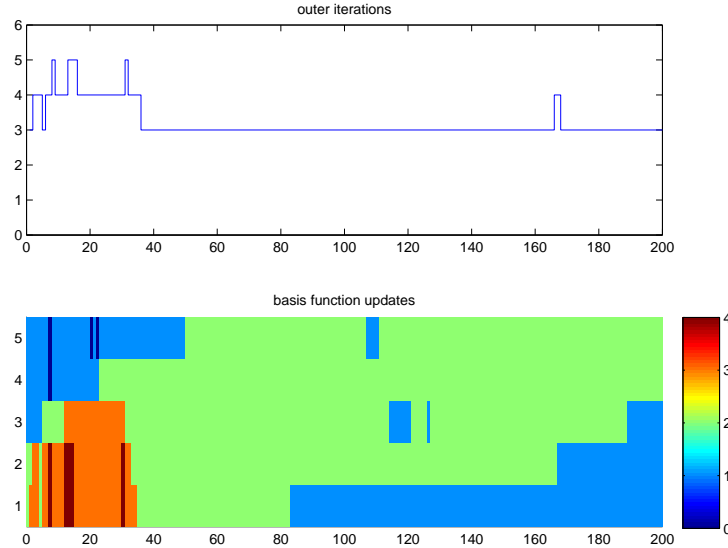
For cases that are strictly compressible in the sense that the divergence of the velocity exceeds a certain lower tolerance in all cells at all times, we can use an entirely different approach. If we have a (good) estimate  $\mathbf{p}^0$  of the fine-scale pressure solution (e.g., from a previous iteration or time step), we can use it to compute a set of *compressible basis functions*. To this end, we manipulate the mixed discrete system (here without sub- and superscripts) and use  $\mathbf{p}^0$  to evaluate the new right-hand side

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \psi \\ -\phi \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} + \mathbf{P}\mathbf{p}^0 \end{bmatrix}. \quad (20)$$

In other words, we choose the weight function  $\mathbf{w} \propto \mathbf{P}\mathbf{p} + \mathbf{g}$  in (12) so that it depends on the solution. The estimated solution is also used to set *global* boundary conditions for the local basis functions. Notice that this strategy only works if  $|\int_{B_i} (Pp + g)(\mathbf{x}) d\mathbf{x}| \gg 0$ .

In this particular setup, we can reduce the mixed system to a system for the unknown fluxes only. The matrix  $\mathbf{P}$  is diagonal and if the flow is compressible in all cells, it is also invertible. Hence, we can use the second equation in the mixed system to eliminate the unknown pressures

$$\mathbf{C}^\top \mathbf{v} - \mathbf{P}\mathbf{p} = \mathbf{g} \implies \mathbf{p} = \mathbf{P}^{-1} (\mathbf{C}^\top \mathbf{v} - \mathbf{g})$$



**Figure 8** Work to reduce the fine-scale residual in the cMsMFE method below  $10^{-5}$  for the 1D tracer-gas example. The upper plot shows the number of outer iterations and the lower plot shows the number of basis-function updates (time step on the x axis and basis function number on the y axis)

By inserting this expression into the first equation, the mixed system can be reduced to a system for the flux only

$$(\mathbf{B} - \mathbf{C}\mathbf{P}^{-1}\mathbf{C}^T)\mathbf{v} = \mathbf{f} - \mathbf{C}\mathbf{P}^{-1}\mathbf{g}.$$

This leads to an iterative method consisting of the following steps:

1. Solve the multiscale *flux* system

$$\begin{aligned} \Psi^T(\mathbf{B} - \mathbf{C}\mathbf{P}_v^{-1}\mathbf{C}^T)\Psi\mathbf{v}_c^{v+1} &= \Psi^T(\mathbf{f} - \mathbf{C}\mathbf{P}^{-1}\mathbf{g})_v \\ \mathbf{p}_{ms}^{v+1} &= \mathbf{P}_v^{-1}(\mathbf{C}^T\Psi\mathbf{v}_c^{v+1} - \mathbf{g}_v) \end{aligned} \quad (21)$$

using  $\Psi\mathbf{v}_c^v + \hat{\mathbf{v}}$  and  $\mathbf{p}_{ms}^v + \hat{\mathbf{p}}^v$  to set boundary conditions.

2. Solve the residual problem

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{P}_{v^*} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{v}}^{v+1} \\ -\hat{\mathbf{p}}^{v+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{v^*} - \mathbf{B}\Psi\mathbf{v}_c^{v+1} + \mathbf{C}\mathbf{p}_{ms}^{v+1} \\ \mathbf{g}_{v^*} - \mathbf{C}^T\Psi\mathbf{v}_c^{v+1} + \mathbf{P}_{v^*}\mathbf{p}_{ms}^{v+1} \end{bmatrix} \quad (22)$$

using (non)overlapping domain decomposition.

To initialize the process, we can either use a fine-scale solution or the *elliptic* basis functions. Basis functions are only updated if the local residuals exceed a prescribed threshold. We will refer to the resulting method as cMsMFE.

**Example 4** We revisit the tracer gas injection from Example 1. Figure 8 shows the work required to reduce the fine-scale residual below  $10^{-5}$ . As for iMsMFE, the number of outer iterations is quite low for all time steps. The results are not directly comparable since they use different convergence criteria. In the lower plot of the figure we observe how the updates of basis functions follows the propagation of the pressure front.

### A bootstrapping multiscale mixed finite-element method

The two approaches discussed above have used a single flux basis function for each interface in the coarse grid and compensated for changes in the local (or global) flow patterns by computing local residuals or updating basis function dynamically throughout the simulation. Another approach is to use a

fixed or an expanding dictionary of basis functions. To this end, we assume that we have a series of boundary conditions  $v_{ij}^1, \dots, v_{ij}^n$  for each coarse interface. The set of boundary conditions can be obtained in different ways:

- By solving a sequence of global (single-phase) flow problems using generic boundary conditions (e.g., flow in the axial directions, etc).
- Extracted from a previous simulation, for instance if the forward simulation is repeated multiple times with small changes in parameters as in a history-matching or optimization workflows. Then it may be useful to combine the multiscale basis functions with a local proper orthogonal decomposition (POD) as discussed in [14].
- Using a bootstrapping method in which an expanding dictionary of basis functions is built from previous time (and iteration) steps.

To form a dictionary of basis functions for an interface, we collect the corresponding boundary conditions  $v_{ij}^k$  as columns in a local matrix  $\mathbf{N}_{ij}$  and then perform a singular value decomposition (SVD)

$$\mathbf{N}_{ij} = \mathbf{U}_{ij} \Sigma_{ij} \mathbf{V}_{ij}^T = \sum_{\ell=1}^n \mathbf{u}_{ij}^{\ell} \sigma_{ij}^{\ell} (\mathbf{v}_{ij}^{\ell})^T \quad (23)$$

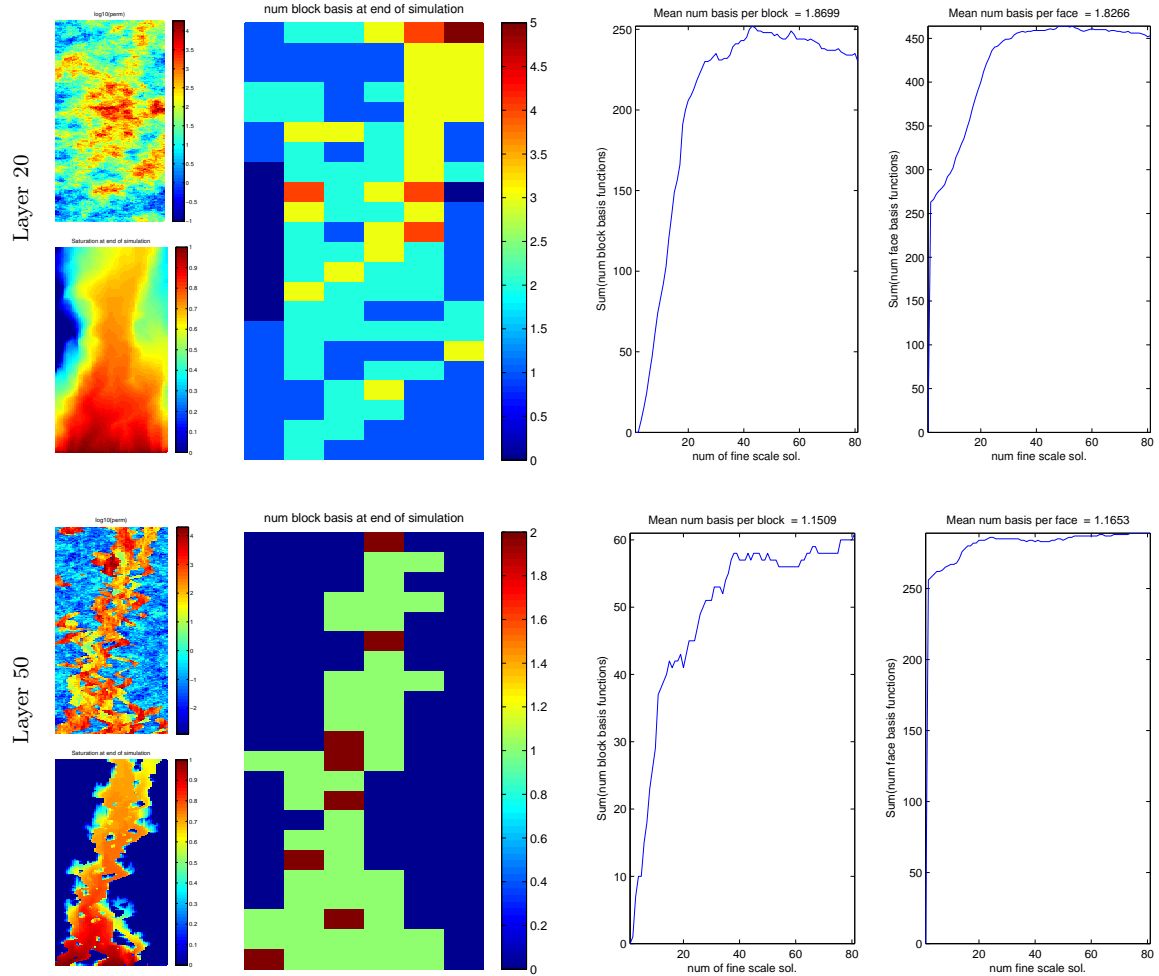
where  $\mathbf{U}_{ij}$  and  $\mathbf{V}_{ij}$  are unitary matrices and  $\Sigma_{ij}$  is a diagonal matrix containing the singular values  $\sigma_{ij}^{\ell}$  in descending order. For a given prescribed tolerance  $\varepsilon$ , we let  $m$  be the smallest number so that either  $|\sigma_{ij}^{m+1}| < \varepsilon$  or  $\sum_{\ell=m+1}^n \sigma_{ij}^{\ell} < \varepsilon$ . Then, we use  $\mathbf{u}_{ij}^1$  as global boundary conditions to construct basis functions  $\tilde{\psi}_{ij}^1$  from (7) and (8). The remaining  $m-1$  snapshots  $\mathbf{u}_{ij}^2, \dots, \mathbf{u}_{ij}^m$  are shifted to have zero mean

$$\mathbf{u}_{ij}^{\ell} - \frac{\int_{\Gamma_{ij}} \mathbf{u}_{ij}^{\ell}}{\int_{\Gamma_{ij}} \mathbf{u}_{ij}^1} \mathbf{u}_{ij}^1$$

and then used to compute divergence-free basis functions  $\tilde{\psi}_{ij}^{\ell}$ ,  $\ell = 2, \dots, m$  from (9) and (10). The corresponding set of basis functions now spans the given snapshots to prescribed accuracy over the coarse grid interfaces, and accordingly, the *remaining* snapshot-span is localized to each coarse block. This span can then efficiently be captured by a set of block-local PODs to obtain an additional block-basis (see Krogstad [14] for details). In this setup, the number of basis functions can vary from one interface to another and also expand with time. Here, one idea is to adaptively *enrich* the basis where needed (considering the fine-scale residual) through the use of local correction functions.

Even before such a method is implemented, we can say something about its potential by analyzing the output from a fine-scale simulation. In the next example, we therefore compute the number of basis functions needed to span the dynamical solution in two heterogeneous reservoirs. To this end, we run fine-scale simulations, save the flux at each time step, and then look at how many basis functions we need to span the fine-scale flux solutions to a certain tolerance level. This will give an indication on how often the multiscale basis would need to be enriched during a multiscale simulation (if no prior knowledge was available).

**Example 5 (SPE 10 in 2D)** We consider two layers of the SPE 10 benchmark model [9] that each consist of  $60 \times 220$  cells: Layer 20 is rather smooth, while Layer 50 is fluvial. The flow is driven by a pressure drop of 200 bar from bottom to top, and water is injected into oil. Both fluids are assumed to be incompressible with quadratic relative permeability curves and a mobility ratio of 10. We simulate 8000 days and sample 80 snapshots of the fine-scale solution that are used to form a basis-function dictionary for a  $6 \times 22$  coarse grid. For each snapshot  $i$ , we calculate the number of basis functions needed to span fine-scale solutions  $\{1, 2, \dots, i\}$  to a given relative tolerance. Figure 9 and Table 2 show that less basis functions are needed for Layer 50, where the flow is localized to the channel-like structure. Moreover, the majority of the basis functions are added during the early time-steps, while the number of basis



**Figure 9** The plots to the left show permeability and saturation profile at the end of simulation for Layers 20 and 50 of the SPE 10 data set. The middle plots show distribution of block basis functions at the end of simulation. The plots to the right show the total number of face and block basis functions needed to span the fine-scale flux solution  $\{1, 2, \dots, i\}$  to tolerance  $10^{-2}$ .

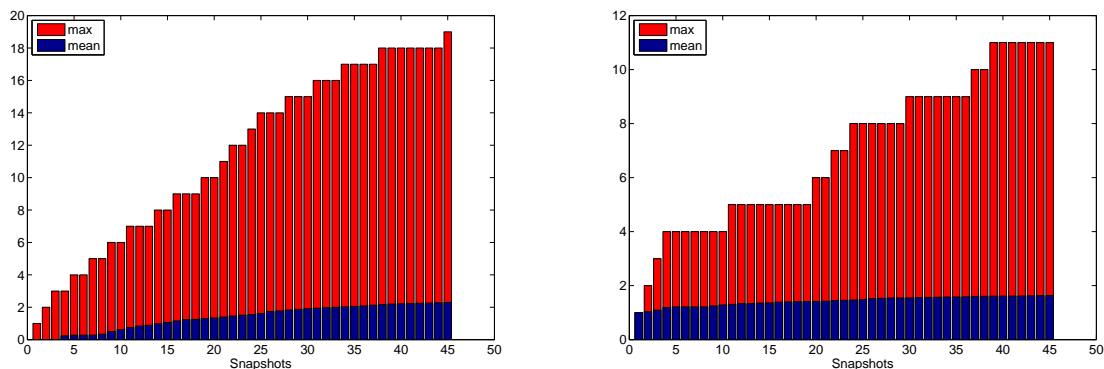
functions at later times stays more or less constant. A significant computational effort would therefore be needed early in a multiscale simulation, while little effort can be spent on local computations later in the simulation. Extensive numerical experiments conducted previously show that the MsMFE method generally is well suited for capturing heterogeneity effects having long correlation lengths.

The previous example demonstrated how inclusion of a few extra basis functions may significantly improve MsMFE's ability to resolve heterogeneity and other effects such as capillary pressure. In the next example we investigate this further by considering output from the simulation of a real field model.

**Example 6 (Analysis of flux-field from a black-oil simulation)** To further evaluate the potential of the previous approach, we here consider output from a 3-phase black-oil simulation of a real field from the Norwegian Sea. We compute and collect the flux fields from 45 snapshots throughout the simulation, and compute a basis dictionary that spans the snapshots to a relative tolerance of  $1e-3$  resulting from a coarse  $10 \times 4 \times 5$  partitioning. The mean and maximum number of basis functions needed per block and face, respectively, are depicted in Figure 10. It is worth noting that even though a few blocks/faces require a high number of basis functions, the mean remains small, and accordingly the coarse dimension remains low. In the left plot in Figure 11, we have visualized the distribution of block basis functions required to span all the 45 snapshots. The right plot shows the initial gas saturation. We observe that

**Table 2** Mean number of block / face basis functions needed to achieve a prescribed relative accuracy on layers of SPE 10 at the end of simulation with coarse grid of  $6 \times 22$ .

Accuracy	Layer 20	Layer 50
$10^{-2}$	1.87 / 1.83	1.15 / 1.17
$10^{-4}$	11.27 / 6.42	5.90 / 2.69
$10^{-6}$	20.85 / 9.68	10.55 / 5.49

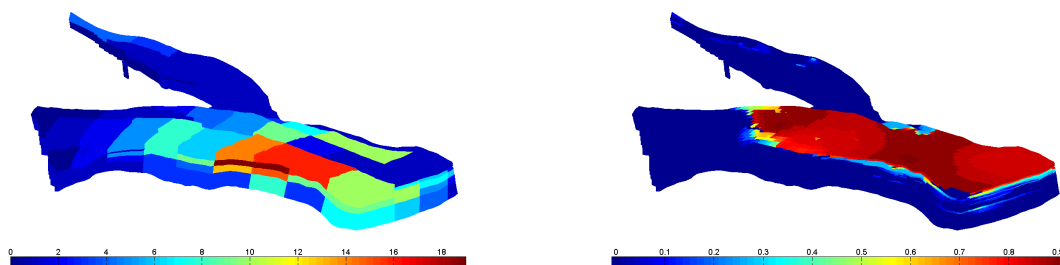


**Figure 10** Maximum and mean number of basis functions for coarse blocks(left) and faces (right) needed to span the black-oil simulation output snapshots

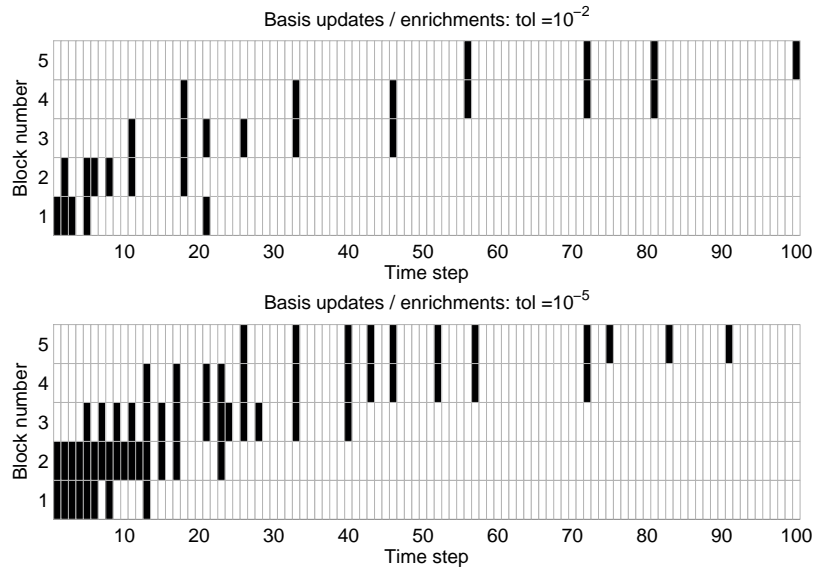
the block basis density is highest close to the gas cap, and in particular in the zone where the gas and oil phases interact. It follows that for an accurate multiscale simulation of such a model, many of the basis functions could be reused, but frequent updates/enrichments would be needed where different phases interact.

In the next example, we will demonstrate how to use bootstrapping to dynamically enrich the local dictionaries of basis functions that resolve compressibility effects. The resulting method is referred to as bMsMFE.

**Example 7** We revisit the tracer gas injection from Example 1. Figure 12 depicts basis updates and enrichments needed for fine-scale residual tolerances of  $10^{-2}$  and  $10^{-5}$ . In this simplified setup, we perform a fine-scale solve every time the residual exceeds the tolerance, and extract a new basis function (through a scaling and an orthogonalization to existing basis functions). It is evident that by allowing multiple basis functions per face/block we drastically reduce the need for updates compared to Exam-



**Figure 11** Number of basis functions required per block to span all snapshots for the black-oil simulation (left), and initial gas saturation (right)



**Figure 12** Basis updates/enrichments needed for fine-scale residual to be less than  $10^{-2}$  (upper) and  $10^{-5}$  (lower). Dark patches indicate a new basis function is added for the corresponding time-step and coarse block.

ple 4. In particular for the case with tolerance  $10^{-2}$ , which results in a pointwise pressure discrepancy less than 0.1% for all time-steps, updates are only needed in 16 of the 100 time-steps. The remaining time-steps are solved using the multiscale system only.

Using fine-scale solves to enrich the basis dictionary will generally not be efficient. Instead, one could use correction functions as in the iMsMFE method. As presented above, the bootstrapping approach attempts to extract and utilize as much information as possible from previous computations. To keep the coarse system size low, one should limit the number of basis functions per face/block. A relevant approach was presented in [7] in which a few of the previous time steps were used to compute a POD-basis that in turn was used to construct a preconditioner for the (full) non-linear system.

### Comparison and discussion of methods

The standard MsMFE method (without corrections) is the most computationally efficient approach and will typically give decent results for a single pressure solve. That is, although the method may not necessarily provide good pointwise resolution of pressure and fluxes, it will accurately and efficiently capture the evolution of saturation profiles and global flow responses like water-cuts, etc, in particularly for incompressible two-phase flow. For compressible cases, on the other hand, the method gives inconsistencies between flux and pressure that may cause serious problems when coupling with a standard fine-scale transport solver.

The iMsMFE method converges to the fine-scale solution for all the cases we have been able to test. The main problem with the iMsMFE method is that the elliptic basis functions only reduce the fine-scale residual to a certain level and then leaves the rest of the work to an inefficient domain-decomposition corrector. As a result, the number of outer iterations tend to increase non-linearly with decreasing tolerance. Accordingly, the current approach does not appear to be a good alternative to state-of-the-art AMG solvers for solving the fine-scale problem (perhaps except for the obvious parallelism inherent in the computation of basis and correction functions). On the other hand, the residual is considerably reduced by a single iteration in the method and iMsMFE may therefore have a significant potential for cases where a very small fine-scale residual is not required.

The bMsMFE method is very flexible in the sense that it uses multiple basis functions per coarse block or interface. Even though the method has not been extensively tested, analysis shows that it has a good potential. In particular, the bMsMFE method is able to incorporate and *reuse* previous computations (from earlier time-steps and/or simulations) to a much larger degree than the iMsMFE method. Sometimes, however, the method requires an inordinately large number of basis functions to reduce the fine-scale residual below a prescribed tolerance. Moreover, some of the accumulated basis functions may become outdated as pressure signals evolve and thus contribute little to the solution. To this end, one should develop efficient methods to dynamically trim the basis function dictionaries.

Further research is needed to address the full potential of the methods. One potential approach is to combine the iterative and bootstrapping ideas: use bMsMFE as a predictor to significantly reduce the fine-scale residual, and then use correction functions for iterative improvement. Another question is how to combine the 'incompressible' basis functions used for the iMsMFE method with the 'compressible' basis functions used for the cMsMFE method to give an overall method that may work also when  $\mathbf{P}$  is not strictly invertible, e.g., if some cells have zero compressibility. Moreover, local-global approaches should be investigated for parabolic problems, both for iMsMFE and bMsMFE.

Finally, the need to enforce a very small fine-scale residual may diminish if one can develop more robust strategies for computing transport. Here, there are several approaches that could be investigated. Some simulators first solve pressure and fluxes, then evolve saturations, and finally correct the mass balance. Likewise, in streamline simulation one first solves a pressure equation to trace streamlines and then compute flow and transport simultaneously in 1D along each streamline. It should also be possible to develop coarse-scale transport solvers that do not require full fine-scale consistency between fluxes and pressures.

## 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 (electronic).
- [2] Aarnes, J.E., Kippe, V. and Lie, K.A. [2005] Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels. *Adv. Water Resour.*, **28**(3), 257–271.
- [3] Aarnes, J.E., Krogstad, S. and Lie, K.A. [2006] A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Model. Simul.*, **5**(2), 337–363 (electronic).
- [4] Aarnes, J.E., Krogstad, S. and Lie, K.A. [2008] Multiscale mixed/mimetic methods on corner-point grids. *Comput. Geosci.*, **12**(3), 297–315, doi:10.1007/s10596-007-9072-8.
- [5] Alpak, F.O., Pal, M. and Lie, K.A. [2012] A multiscale method for modeling flow in stratigraphically complex reservoirs. *SPE J.*, in press.
- [6] Arbogast, T. and Bryant, S.L. [2002] A two-scale numerical subgrid technique for waterflood simulations. *SPE J.*, **7**(4), 446–457.
- [7] Astrid, P., Papaioannou, G., Vink, J. and Jansen, J. [2011] Pressure preconditioning using proper orthogonal decomposition. *2011 SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21-23 February 2011*, doi:10.2118/141922-MS.
- [8] Chen, Z. and Hou, T. [2003] A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Math. Comp.*, **72**, 541–576.
- [9] Christie, M.A. and Blunt, M.J. [2001] Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Eval. Eng.*, **4**, 308–317, url: <http://www.spe.org/csp/>.
- [10] Hajibeygi, H., Bonfigli, G., Hesse, M.A. and Jenny, P. [2008] Iterative multiscale finite-volume method. *J. Comput. Phys.*, **227**(19), 8604–8621, ISSN 0021-9991, doi:10.1016/j.jcp.2008.06.013.
- [11] Hajibeygi, H. and Jenny, P. [2011] Adaptive iterative multiscale finite volume method. *J. Comput. Phys.*, **230**(3), 628–643, ISSN 0021-9991, doi:10.1016/j.jcp.2010.10.009.
- [12] Jenny, P., Lee, S.H. and Tchelepi, H.A. [2003] Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *J. Comput. Phys.*, **187**, 47–67.
- [13] Kippe, V., Aarnes, J.E. and Lie, K.A. [2008] A comparison of multiscale methods for elliptic problems in porous media flow. *Comput. Geosci.*, **12**(3), 377–398, doi:10.1007/s10596-007-9074-6.
- [14] Krogstad, S. [2011] A sparse basis POD for model reduction of multiphase compressible flow. *2011 SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21-23 February 2011*, doi:10.2118/141973-MS.
- [15] Krogstad, S., Lie, K.A., Nilsen, H.M., Natvig, J.R., Skaflestad, B. and Aarnes, J.E. [2009] A multiscale mixed finite-element solver for three-phase black-oil flow. *SPE Reservoir Simulation Symposium, The Wood-*

- lands, TX, USA, 2–4 February 2009, doi:10.2118/118993-MS.
- [16] Lie, K., Krogstad, S., Ligaarden, I., Natvig, J., Nilsen, H. and Skaflestad, B. [2012] Open-source MATLAB implementation of consistent discretisations on complex grids. *Comput. Geosci.*, **16**, 297–322, ISSN 1420-0597, 10.1007/s10596-011-9244-4.
  - [17] Lunati, I. and Jenny, P. [2006] Multiscale finite-volume method for compressible multiphase flow in porous media. *J. Comput. Phys.*, **216**(2), 616–636.
  - [18] Manzocchi, T. et al. [2008] Sensitivity of the impact of geological uncertainty on production from faulted and unfaulted shallow-marine oil reservoirs: objectives and methods. *Petrol. Geosci.*, **14**(1), 3–15.
  - [19] Møyner, O. [2012] *Multiscale finite-volume methods on unstructured grids*. Master's thesis, Norwegian University of Science and Technology.
  - [20] Zhou, H. and Tchelepi, H. [2008] Operator-based multiscale method for compressible flow. *SPE J.*, **13**(2), 267–273.