



**SPE 173265-MS**

## **A Multiscale Method Based on Restriction-Smoothed Basis Functions Suitable for General Grids in High Contrast Media**

Olav Møyner, Knut-Andreas Lie, SPE, SINTEF ICT.

Copyright 2015, Society of Petroleum Engineers

This paper was prepared for presentation at the SPE Reservoir Simulation Symposium held in Houston, Texas, USA, 23-25 February 2015.

This paper was selected for presentation by an SPE program committee following review of information contained in an abstract submitted by the author(s). Contents of the paper have not been reviewed by the Society of Petroleum Engineers and are subject to correction by the author(s). The material does not necessarily reflect any position of the Society of Petroleum Engineers, its officers, or members. Electronic reproduction, distribution, or storage of any part of this paper without the written consent of the Society of Petroleum Engineers is prohibited. Permission to reproduce in print is restricted to an abstract of not more than 300 words; illustrations may not be copied. The abstract must contain conspicuous acknowledgment of SPE copyright.

### **Abstract**

A wide variety of multiscale methods have been proposed in the literature to improve simulation runtimes and provide better scaling to large models. With a few notable exceptions, the methods proposed so far are mostly limited to structured grids. We present a new multiscale restricted-smoothed basis (MsRSB) method that is designed to be applicable to stratigraphic and fully unstructured grids. Like many other multiscale methods, it is based on a coarse partition of an underlying fine grid with a set of prolongation operators (also called multiscale basis functions) that map from unknowns associated with the fine grid cells to unknowns associated with the coarse grid blocks. These mappings are constructed by restricted smoothing: starting from a constant, a localized iterative scheme is applied directly to the fine-scale discretization to give prolongation operators that are consistent with the local properties of the differential operators. The resulting method has three main advantages: First, there are almost no requirements on the geometry and topology of the fine and the coarse grids. Coarse partitions and good prolongation operators can therefore easily be constructed on complex models involving high media contrasts and unstructured cell connections introduced by faults, pinch-outs, erosion, local grid refinement, etc. Moreover, the coarse grid can easily be adapted to features in the geo-cellular model or any precomputed flow field to improve accuracy. Secondly, the method is accurate and robust when compared to existing multiscale methods. In particular, the method does not need to recompute local basis functions to account for transient behavior: dynamic mobility changes are incorporated by continuing previous iterations with a few extra steps. This way, the cost of updating the prolongation operators will be proportional to the amount of change in fluid mobility and one avoids tolerance-based updates. Finally, since the MsRSB method is formulated on top of a cell-centered, conservative, finite-volume method, it is applicable to any flow model in which one can isolate a pressure equation; in the paper, we discuss incompressible two-phase flow and compressible, three-phase, black-oil type models. For our fine-scale discretization, we use the standard two-point flux-approximation scheme, but the method could equally well have been formulated using a multipoint discretization.

Several numerical examples are presented to highlight features of the method. First, we compare the MsRSB method with the multiscale finite-volume (MsFV) method for single-phase flow problems with petrophysical parameters from the SPE 10 benchmark. Then we perform several validation studies using two-phase flow geometry and petrophysical properties from simulation models of two real fields (Gullfaks and Norne), as well as a compressible gas-injection case described by the black-oil equations.

### **Introduction**

The general movement of fluids in a hydrocarbon reservoir is induced by global forces like gravity and pressure differentials. The micro-scale displacement, however, is determined by small-scale flow paths throughout highly heterogeneous porous rocks. Flow modeling therefore needs to take into account processes taking place on a wide range of spatial and temporal scales. The traditional approach to solving such problems is to use upscaling or homogenization techniques to develop effective parameters that represent a sub-scale behavior in an averaged sense so that flow can be simulated on a coarser scale. Such methods have proved to be very effective for problems with scale separation, e.g., as seen in material science. However, porous rocks seldom exhibit clear scale separation and upscaling techniques are therefore not as robust and accurate as one would wish. Effective properties are generally process dependent, and because one needs to assume a specific set of localization conditions to compute effective properties, upscaling techniques tend to only produce reliable results for a limited range of flow scenarios.

In an attempt to overcome some of the limitations of upscaling methods, so-called multiscale discretization methods have been proposed over the past two decades to solve second-order elliptic equations with strongly heterogeneous coefficients (Efendiev and Hou 2009). This includes methods such as the generalized finite-element methods (Babuka et al. 1994), finite-element methods (Hou and Wu 1997), numerical-subgrid upscaling (Arbogast 2002), multiscale mixed finite-element methods (Chen and Hou

2003), multiscale finite-volume methods (Jenny et al. 2003), mortar mixed finite-element methods (Arbogast et al. 2007), and multiscale mimetic methods (Lipnikov et al. 2008), to name a few. The key idea of all these methods is to construct a set of prolongation operators (or basis functions) that map between unknowns associated with the fine geo-cellular grid cells and unknowns on a coarser grid used for dynamic simulation. The prolongation operators are computed numerically by solving localized flow problems, much in the same way as for flow-based upscaling methods, but unlike effective parameters, the multiscale basis functions have sub-scale resolution. The result is that local fine-scale variations can be systematically and correctly accounted for when constructing a reduced coarse-scale problem to study the macro-scale displacement driven by global forces. There is a large body of literature that develops such multiscale methods and studies their mathematical and numerical properties for idealized and simplified problems. To provide value for commercial applications, however, these methods need to be developed so that they can handle the complexity in flow physics and geological description seen in real-life simulation models. Over the past decade, there have been two main developments in this direction, focusing on the multiscale finite-volume (MsFV) method (Jenny et al. 2003) and the multiscale mixed finite-element (MsMFE) method (Chen and Hou 2003; Aarnes 2004).

Research on the MsFV method has mainly focused on extending the method from incompressible flow to realistic flow physics (Lunati and Jenny 2008, 2006; Lee et al. 2008; Zhou and Tchelepi 2008; Hajibeygi and Jenny 2009; Hajibeygi and Tchelepi 2014) and on developing iterative approaches that ensure that the method converges to the correct solution of the underlying fine-scale discretization (Zhou and Tchelepi 2008; Lunati et al. 2011; Zhou and Tchelepi 2012; Wang et al. 2014). However, with one notable exception (Møyner and Lie 2014a), the MsFV method has so far only been studied on Cartesian or structured grids. Such grids are highly desirable in terms of accuracy, efficiency, and robustness of the numerical discretizations and solvers, and modeling approaches used in industry are therefore predominantly structured in a global sense. However, to accurately account for structural features like faults, joints, and deformation bands and stratigraphic characteristics like channels, lobes, clinoforms, and shale/mud drapes, cell geometries will degenerate and unstructured connections will be introduced locally. Similarly, unstructured connections may be introduced by local grid refinement, e.g., in the near-well zones. The challenge in extending the MsFV method to realistic stratigraphic grids, or in the more general sense to grids with fully unstructured topologies, lies in the underlying primal-dual coarse partition. The MsFV method computes basis functions on a dual partition to define transmissibilities in a multi-point coarse-scale discretization, which is combined with another set of flow problems on the primal partition to reconstruct conservative fine-scale fluxes. We have previously demonstrated that compatible primal-dual partitions can be generated for grids with degenerate cells and unstructured topologies (Møyner and Lie 2014a). However, the coarsening process is difficult to automate in a robust manner, and so far our most advanced algorithm is only able to provide semi-structured partitions for a limited range of coarsening factors. It is also well known that highly contrasted media and large anisotropy ratios may introduce strong non-monotonicities that are hard to get rid of in the iterative stages of the method (Wang et al. 2014).

For the MsMFE method, on the other hand, the main focus has been on making the method as geometrically flexible as possible and developing coarsening strategies that semi-automatically adapt to barriers, channels, faults, and wells in a way that ensures good accuracy for a chosen level of coarsening. The resulting method can efficiently predict two-phase flow patterns that are qualitatively correct for highly heterogeneous and geologically complex reservoir models (Aarnes et al. 2008; Natvig et al. 2011; Alpak et al. 2012; Pal et al. 2013). The method has also been extended towards realistic flow physics (Krogstad et al. 2009, 2012), but this has proved difficult to achieve in a fully robust manner because of the inherent assumption of a pressure equation written on mixed form. In a recent work (Møyner and Lie 2014b), we presented a fully algebraic finite-volume framework that combines the best features of the MsFV and MsMFE methods and developed one specific method that mimics a coarse-scale two-point discretization. By using numerically generated partition-of-unity functions to glue together elementary flow solutions associated with interfaces between coarse blocks, the resulting MsTPFA method could robustly handle complex industry-standard grids with high aspect ratios and unstructured connections without significant impact to the solution quality. Herein, we present a much simpler multiscale formulation that also is designed to offer the same robustness and flexibility, but dispense with some of the intricate details of defining and computing the partition-of-unity functions.

The new method constructs mappings based on restricted smoothing: starting from a constant, a localized iterative scheme is applied directly to the fine-scale discretization to give prolongation operators that are consistent with the local properties of the differential operators. A series of numerical experiments show that the new MsRSB method gives highly accurate prolongation operators for a wide variety of block shapes, e.g., including blocks that adapt to complex geological features in real-world models. Moreover, whereas methods like MsFV, MsMFE, and MsTPFA recompute the prolongation operator locally when faced with mobility changes in the underlying grid, the new method just continues the iteration until the operators are sufficiently smooth. This way, the cost of updating the prolongation operator will be proportional to the amount of change in fluid mobility, eschewing the typical tolerance based updates. The formulation is algebraic and can be applied directly to linear systems, possibly in combination with existing multiscale techniques such as local stages and iterative cycles. Through a series of numerical experiments, which include the well-known SPE 10 data and grid and petrophysical properties from two Norwegian oil fields, we validate our new MsRSB method and show that it is robust and efficient for single-phase and multiphase flow models. In particular, we demonstrate that the method can be used to provide one order-of-magnitude speedup compared to a fully-implicit simulator with CPR and algebraic multiscale preconditioner for a compressible water-injection case. Møyner (2014) reports a comparison of the MsFV, MsTPFA, and MsRSB methods used as iterative solvers, which shows that the MsRSB either performs equally well or clearly outperforms the other two methods for the studied test cases.

## Model Problems

The type of multiscale methods discussed herein are designed to provide accurate discretization of second-order elliptic differential operators of the form  $\nabla \cdot \mathbf{K}(\mathbf{x})\nabla$ , where the coefficient  $\mathbf{K}(\mathbf{x})$  may exhibit orders of magnitude variations over short distances and contain short, intermediate, and long-range correlations. The operators primarily determines the pressure distribution, but may also govern temperature in thermal models.

There are two different ways to form pressure equations that contain this elliptic operator. The most straightforward is through a sequential solution procedure, in which flow (pressure and fluxes) and transport (saturation and/or components) are computed in separate steps, and the operator appears as part of an elliptic or a nonlinear parabolic equation for the fluid pressures. In non-compositional simulation it is more common to use a fully implicit discretization in which pressure and saturation (and components) are solved for simultaneously. However, because flow and transport have very different mathematical characteristics, the solution procedure can be accelerated by separating out strongly coupled, pressure-like components and use these to precondition the system. Multiscale methods can be used in both cases, as a standalone pressure solver or to solve the elliptic part of a constrained pressure residual (CPR) preconditioner, e.g., as an alternative to a multigrid linear solver (Cusini et al. 2014). In our experience, however, multiscale methods are most efficient when used as part of a sequential solution strategy as the key feature of multiscale methods is to obtain reasonable velocity fields from approximate pressure solutions, something which is currently not possible in a fully implicit strategy. Thus, a sequential formulation will be the focus of this paper.

**Single phase flow.** To introduce the multiscale method and investigate its spatial approximation properties, it is sufficient to consider an incompressible, single-phase problem, which is modeled by the variable-coefficient Poisson equation,

$$-\nabla \cdot (\mathbf{K}\nabla p(\mathbf{x})) = q(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d, \quad \mathbf{K} \in \mathbb{R}^d \times \mathbb{R}^d, \quad (1)$$

where  $p$  is the fluid pressure,  $\mathbf{K}$  is the permeability, and  $q$  denotes source terms. We discretize this system using a standard finite-volume scheme in which a transmissibility  $T_{ij}$  is associated with the interface between each pair of two cells  $i$  and  $j$  and used to define a two-point flux approximation,  $v_{ij} = -T_{ij}(p_i - p_j)$ . For a Cartesian grid in 3D, this gives the standard seven-point finite-difference stencil. The resulting linear system

$$A\mathbf{p} = \mathbf{q} \quad (2)$$

is weakly diagonally dominant because of volume conservation over all cells.

**Multiphase flow.** Later in the paper we will study several multiphase flow models, ranging from a simple incompressible two-phase flow model without gravity and capillary forces, to a full three-phase, black-oil model with gas dissolved in oil. We therefore start by outlining the model in its most general form and show how to formulate separate flow and transport equations as part of a sequential formulation. The standard black-oil equations describe mass conservation of oil, water, and gas

$$\partial_t(\phi b_o s_o) + \nabla \cdot (b_o \vec{v}_o) - b_o q_o = 0, \quad (3a)$$

$$\partial_t(\phi b_w s_w) + \nabla \cdot (b_w \vec{v}_w) - b_w q_w = 0, \quad (3b)$$

$$\partial_t[\phi(b_g s_g + b_o r_s s_o)] + \nabla \cdot (b_g \vec{v}_g + b_o r_s \vec{v}_o) - (b_g q_g + b_o r_s q_o) = 0. \quad (3c)$$

Here,  $\phi$  is the pore volume while  $s_\alpha$  and  $b_\alpha$  denote the fluid saturation and inverse formation volume factor (ratio between volume at elevated pressure and volume at surface conditions) of phase  $\alpha = w, o, g$ . The phase velocities  $\vec{v}_\alpha$  are given by a multiphase extension of Darcy's law,

$$\vec{v}_\alpha = -\frac{k_{r\alpha}\mathbf{K}}{\mu_\alpha}(\nabla p_\alpha - \rho_\alpha g \nabla z), \quad (4)$$

where  $p_\alpha$  is the phase pressure,  $k_{r\alpha}$  is the relative permeability, and  $\mu_\alpha$  the viscosity of phase  $\alpha$  and  $g$  is the gravity acceleration. The gas is miscible in oil and the gas-oil ratio  $r_s$  for a saturated oil is a function of pressure. The formation volume factor and the viscosity of oil depend on both pressure and the gas-oil ratio. The model consisting of Eqs. 3 and 4 has more unknowns than equations, and we therefore impose a closure relationship for the saturations,  $\sum_\alpha s_\alpha = 1$ , and two relationships for the phase pressures,  $p_o - p_w = p_{cow}$  and  $p_g - p_o = p_{cgo}$ , where the capillary pressures  $p_{cow}$  and  $p_{cgo}$  are known functions of the phase saturations.

*The pressure equation.* To derive a pressure equation from Eq. 3 we start by discretizing in time using the backward Euler method. This gives us the semi-discrete equations,

$$0 = \mathcal{R}_o = \frac{1}{\Delta t} [(\phi b_o s_o)^{n+1} - (\phi b_o s_o)^n] + \nabla \cdot (b_o \vec{v}_o)^{n+1} - (b_o q_o)^{n+1}, \quad (5a)$$

$$0 = \mathcal{R}_w = \frac{1}{\Delta t} [(\phi b_w s_w)^{n+1} - (\phi b_w s_w)^n] + \nabla \cdot (b_w \vec{v}_w)^{n+1} - (b_w q_w)^{n+1}, \quad (5b)$$

$$0 = \mathcal{R}_g = \frac{1}{\Delta t} [(\phi b_g s_g + \phi r_s b_o s_o)^{n+1} - (\phi b_g s_g + \phi r_s b_o s_o)^n] + \nabla \cdot (b_g \vec{v}_g + b_o r_s \vec{v}_o)^{n+1} - (b_g q_g + b_o r_s q_o)^{n+1}. \quad (5c)$$

Next, we assume that there exist factors  $\beta_\alpha$  so that we can obtain an equation on the form,

$$0 = \mathcal{R}_p = \beta_w \mathcal{R}_w + \beta_o \mathcal{R}_o + \beta_g \mathcal{R}_g, \quad (6)$$

where  $\mathcal{R}_p$  does not depend on the saturations at the next time step in the accumulation term. For the black-oil model it is easy to show that these factors are given as

$$\beta_w = \frac{1}{b_w^{n+1}}, \quad \beta_o = \frac{1}{b_o^{n+1}} - \frac{r_s^{n+1}}{b_g^{n+1}}, \quad \beta_g = \frac{1}{b_g^{n+1}}, \quad (7)$$

so that the saturations at the next time step can be eliminated by the closure equation  $\sum_\alpha s_\alpha = 1$ . In the special case of incompressible fluids in an incompressible rock, we end up with a simple pressure equation of the form  $\nabla \cdot \sum_\alpha \vec{v}_\alpha = q$ .

As for the single-phase equation, we use a standard two-point finite-volume scheme for the spatial discretization, extended with upstream weighting of all terms that depend on saturation. (Note, however, that we could equally well have used a multipoint flux-approximation for the spatial discretization.) Once we have the residual equation, we can pick one of the phase pressures as our primary unknown  $p$ , and then solve the nonlinear system of discrete equations using Newton's method in a standard manner,

$$p^{n+1} = p^n - J_p(p^n)^{-1} \mathcal{R}_p(p^n) \quad (8)$$

where  $J_p$  refers to the Jacobian  $d\mathcal{R}_p/dp$  of the residual equation  $\mathcal{R}_p$ .

*Transport equations.* To derive transport equations for updating the fluid saturations, we start by assuming that the pressure iteration Eq. 8 has converged to the desired tolerance. If we define the total flux  $\vec{v}_T$  at reservoir conditions based on the pressure equation and insert the resulting expression for  $\nabla p$  into each phase flux we obtain

$$\vec{v}_T = \sum_\alpha \vec{v}_\alpha, \quad \vec{v}_\alpha = f_\alpha \left[ \vec{v}_T + \mathbf{K} \sum_\nu \lambda_\nu^f (\rho_\alpha^f g \nabla z - \rho_\nu^f g \nabla z) \right], \quad f_\alpha = \frac{\lambda_\alpha^f}{\sum_\nu \lambda_\nu^f}. \quad (9)$$

Here, the densities at the interfaces  $\rho_\alpha^f$  are averaged between the two cells on opposite sides of the interface and are defined so that the oil density also accounts for the dissolved gas,

$$\rho_w = b_w \rho_w^s, \quad \rho_o = b_o (\rho_o^s + r_s \rho_g^s), \quad \rho_g = b_g \rho_g^s. \quad (10)$$

For the mobilities on the faces  $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$  we use potential ordering to determine the upstream weighting (Brenier and Jaffré 1991) which will for most situations coincide with the upstream weighting used in the pressure solver. To make the expressions easier to read, we have tacitly assumed that capillary forces are negligible so that there is only a single unique pressure. In general, these terms appear along with the gravity contributions in the potential differences, and are also included in the implementation.

Now that we have expressions for the fluxes, we can solve two of the three conservation equations. Typically, we solve the oil and gas equations to ensure that the hydrocarbon masses are conserved throughout the simulation and let the water phase fill up the remaining pore volume. In the classical IMPES method, the saturations are updated using a single step of an explicit scheme. Herein, we use an implicit solver to allow for longer time steps. That is, we use Newton's method for Eqs. 5a and 5c with fluxes provided by Eq. 9 to obtain saturations for the next time step. If necessary, one can also iterate the pressure and transport steps to ensure that the residual of the combined discrete system is below a prescribed threshold. However, for the cases considered later in the paper, we have not found this necessary.

## Multiscale Formulation

As explained above, our definition of a multiscale method starts from a fine grid,  $\Omega = \{\Omega_i\}_{i=1}^n$ , and a coarse partition that defines a coarse grid,  $\bar{\Omega} = \{\bar{\Omega}_i\}_{i=1}^m$ , so that each cell in  $\Omega$  belongs to only one block in  $\bar{\Omega}$ . We then define a numerical prolongation operator,  $P : \bar{\Omega} \rightarrow \Omega$ , that maps quantities associated with the coarse blocks to quantities associated with the fine cells. Likewise, we define a restriction operator as the analogous map going the other way  $R : \Omega \rightarrow \bar{\Omega}$ . In the implementation, these operators will be represented as sparse matrices of size  $n \times m$  and  $m \times n$  respectively. If we now let  $\mathbf{p}_c$  denote a pressure computed on the coarse grid, we can find a fine-scale approximate pressure  $\mathbf{p}_f$  by the use of the prolongation operator,

$$\mathbf{p}_f = P\mathbf{p}_c. \quad (11)$$

In general this will not solve Eq. 1 exactly no matter how accurate the coarse pressure is; all we can hope for is to compute a good and accurate approximation more efficiently than solving Eq. 1 directly on  $\Omega$ .

**Coarse system.** To derive a linear system for  $\mathbf{p}_c$  on the coarse grid, we insert the fine-scale approximation in Eq. 11 into Eq. 2 and apply the restriction operator,

$$R(A(P\mathbf{p}_c)) = (RAP)\mathbf{p}_c = A_c\mathbf{p}_c = R\mathbf{q} = \mathbf{q}_c. \quad (12)$$

The physical interpretation of this system depends on the restriction operator used. In the literature, the variants used are typically either a control volume summation operator or a Galerkin operator, i.e.,

$$(R_{cv})_{ji} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \bar{\Omega}_j, \\ 0, & \text{otherwise,} \end{cases} \quad \text{or} \quad R_G = P^T. \quad (13)$$

Since we are not focusing on iterative performance here, it is naturally to focus on the control volume operator, which corresponds to setting the transmissibility from coarse block  $\bar{\Omega}_i$  into coarse block  $\bar{\Omega}_j$  as the sum of the fluxes induced by the prolongation operator defined in block  $i$  across the interface  $\partial\bar{\Omega}_{ij}$ . This is the same restriction operator that is used in the classical MsFV method (Jenny et al. 2003).

The pressure obtained by prolongating the solution of Eq. 12 back to the fine scale is generally not an exact solution of Eq. 2. The coarse-scale system imposes mass balance on the coarse scale and from the pressure solution we can easily compute fluxes that are conservative on the coarse grid. However, to get fluxes that are conservative also on the subscale, we can solve an additional local problem with the conservative, coarse-scale fluxes imposed as Neumann boundary conditions. This way, it is possible to solve fine-scale transport to a high accuracy without the exact pressure being known. The disadvantage of using this formulation is that we risk producing negative coarse-scale transmissibilities, which may lead to unphysical solutions having non-monotone pressure values that violate the maximum principle (Møyner and Lie 2014a).

To define a specific multiscale method, we must also describe in detail how to construct the prolongation operator  $P$ . As in most other multiscale methods, we will construct  $P$  by piecing together a set of localized functions. However, before we can describe these so-called basis functions, we must provide more details about the coarse grid.

**Coarse grids and interaction regions.** Most multiscale formulations rely on a combination of a primal coarse grid and an auxiliary spatial characterization to obtain localized functions and piece them together to form the prolongation operator. For the MsFV method, this additional mechanism is a dual coarse grid, whereas for the MsTPFA method (Møyner and Lie 2014b), the additional mechanism is a set of localized partition-of-unity functions that are computed numerically over the fine grid. Previous research has shown that the choice of the primal grid (and the auxiliary spatial characterization) can have a pronounced impact on the solution quality. We therefore want a formulation that is as flexible as possible and enables basis functions to be computed for fairly general coarse grids. Even for problems posed on a Cartesian mesh without faults or pinched cells, the solution can be greatly improved with the use of coarse grids that adapt to local features in the permeability field or flow patterns (Møyner and Lie 2014a,b; Møyner 2014).

As mentioned above, the primal coarse grid is defined through a partition vector that has a single unique indicator value per fine cell, so that cells with the same indicator value are agglomerated into coarse blocks. For structured grids, these indicators can simply be based on counting, forming logically hexahedral blocks on the coarse scale, or they may be formed using a wide variety of graph-based partitioning algorithms. Let  $F$  be the index set of fine cells and let  $C_j$  be the set of fine-scale indices corresponding to coarse block number  $j$ ,

$$C_j \subseteq F, \quad C_j \cap C_i = \emptyset \quad \forall \quad i \neq j, \quad i, j \in [1, m], \quad |F| = n. \quad (14)$$

Once we have defined a coarse grid, we must define the *interaction regions*. Doing this determines the support of a basis function, so if we let  $I_j$  denote the set of all points contained in the interaction region for coarse block  $\bar{\Omega}_j$  and  $P_j$  the basis function of coarse block  $j$  this implies that

$$P_j(\mathbf{x}) > 0, \quad \mathbf{x} \in I_j \quad P_j(\mathbf{x}) = 0 \text{ otherwise.} \quad (15)$$

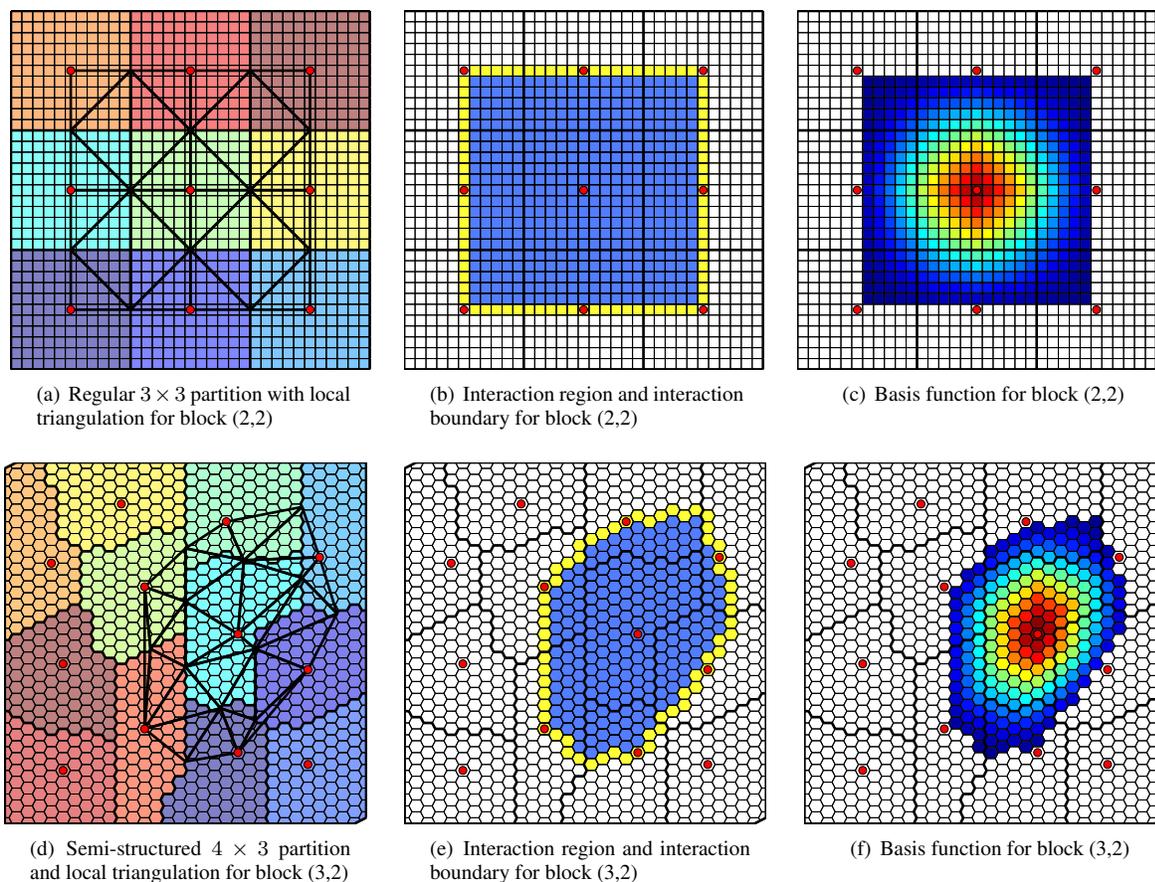
Likewise, we define the *interaction boundary*  $B_j$ , which is the index set of cells that are topological neighbors to the interaction region  $I_j$ , but are not themselves contained in it. We also define the *center* of a coarse block  $\bar{\Omega}_j$  as  $\mathbf{x}_j^c$ . This is a single point, which in many cases *may* coincide with the centroid of the coarse block. For convenience, we also define the global boundary  $G$  as the index set of all fine cells that are part of the interaction boundary of one or more coarse blocks,

$$G = B_1 \cup B_2 \cup \dots \cup B_{m-1} \cup B_m. \quad (16)$$

Finally, for each fine cell that is member of the global boundary, we define  $H_i$  to be the set of indices of the coarse support regions the cell belongs to,

$$H_i = \{j \mid i \in I_j, i \in G\}. \quad (17)$$

To make this notation easier to visualize, we refer to **Fig. 1**, which shows a regular partition for a uniform Cartesian grid and a semi-structured partition for a PEBI grid. The figure also shows the construction of interaction regions, interaction boundaries, and basis functions.



**Figure 1—Primal coarse grid and construction of a local interaction region with an associated basis function for a uniform Cartesian grid in the upper row and for a perpendicular bisector (PEBI) grid in the lower row.**

Careful numerical experiments have shown that rather than setting the block centroid as the block center, we should choose the geometric median of the fine-scale faces that bound the block. For regular coarse grids, the two choices coincide, but the geometric median gives basis functions of better quality for coarse partitions with large variation in block sizes and shapes. **Fig. 2** shows an exaggerated example with a large block neighboring ten small blocks that each consist of a single fine-scale cell. The basis function computed with the centroid defined as block center contains a large constant region. Using the geometric mean instead reduces the constant region and improves the approximation quality of the prolongation operator significantly.

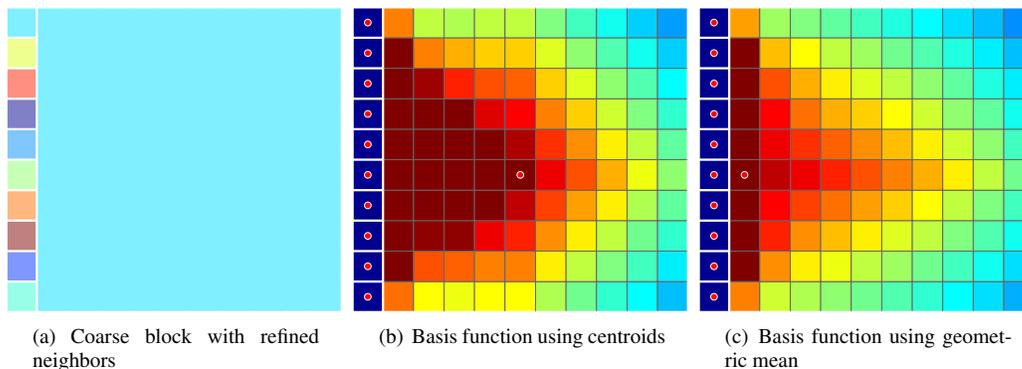
To define the interaction region of block number  $i$ , we select all blocks that share a coarse face with block  $i$  and create a local triangulation based on the block centers and the centroids of all coarse faces that are shared by any two of these blocks, see Figs. 1(a) and 1(d). The interaction region is then defined as all cells whose centroid lies within the triangulation, and the interaction boundary is defined as all cells that share at least one face with cells in the interaction region. Because all these relations can be produced using only topology information, block centers, and face centroids, the implementation is the same regardless of whether we identify interaction regions in a two-dimensional Cartesian grid or in a complex field model in 3D.

**Construction of basis functions.** Most multiscale methods rely on the solution of local problems to produce the basis functions that form the prolongation operator. These local problems are typically defined as some subset of the global problem with alternate boundary conditions imposed to capture the local features. Herein, we will deviate from this and instead construct the basis functions using an iterative process. A similar approach can be found in some multigrid methods that employ a single step of a smoother applied directly to a simple prolongation operator to reduced local error (Jacobi interpolation), see (Vanek et al. 1994).

The basis functions are initially defined to be constant values for each coarse block,

$$P_{ij}^0 = \begin{cases} 1 & \text{if } i \in C_j \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

Nothing is preventing us from choosing some more intricate initial guess for the prolongation operator, but constant functions are convenient because they are trivial to construct and automatically provide partition of unity.

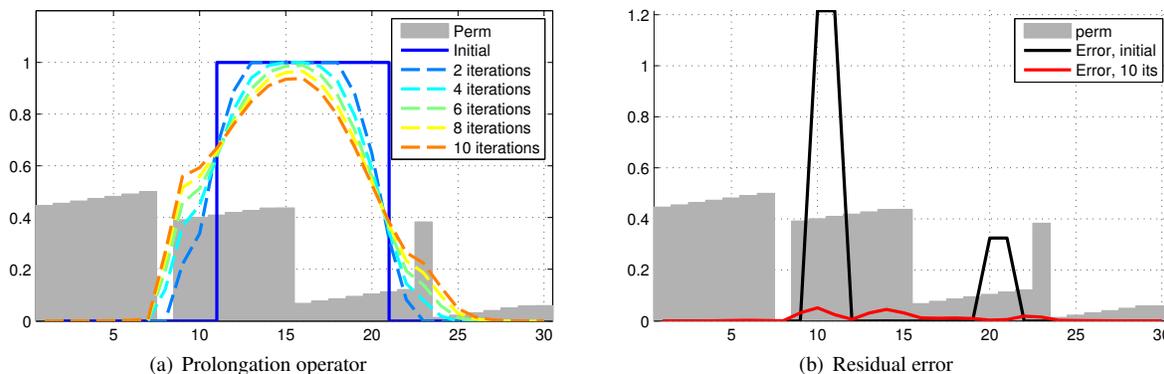


**Figure 2—The choice made for the definition of the center points will affect the quality of the prolongation operator. This example shows a coarse block where the neighboring blocks consist of a single cell. The prolongation operator becomes smoother when the block center is defined as the geometric median of the face centroids rather than as the block centroid.**

We define the local smoother as,

$$P_j^{n+1} = P_j^n - \omega D^{-1} A P_j^n, \quad (19)$$

where  $D$  is the diagonal entries of  $A$ , our (weakly) diagonally dominant system matrix, and  $\omega$  a relaxation factor which we typically set to  $1/2$  or  $2/3$ . By altering the prolongation operator, we seek to make it algebraically smooth, i.e., reduce  $\|AP\|_\infty$  by as much as possible. This means that the residual error in the prolongation operator should be relatively smooth. **Fig. 3** shows a simple 1D example of how the operator and the associated error change as more iterations are applied. Because each iteration modifies cell values based on the topological neighbors, we can see that the support of the basis functions will eventually grow to cover the entire domain. To avoid this, we use our already defined interaction regions and boundaries to limit the updates.



**Figure 3—Illustration of how the iterative procedure gradually smooths the basis functions until they have very low residual error.**

Roughly speaking, the basis construction determines an increment for each basis function based on the local error and modifies this increment to limit the support to be within the interaction regions. This update is also used to determine convergence of the basis construction procedure.

1. Apply smoother to find the increment of the discrete basis function,

$$\hat{\mathbf{d}}_j = -\omega D^{-1} A P_j^n. \quad (20)$$

2. Modify the update to avoid stencil growth outside of the interaction region and preserve partition of unity,

$$d_{ij} = \begin{cases} \frac{\hat{d}_{ij} - P_{ij} \sum_{k \in H_j} \hat{d}_{ik}}{1 + \sum_{k \in H_j} \hat{d}_{ik}}, & i \in I_j, i \in G, \\ \hat{d}_{ij}, & i \in I_j, i \notin G, \\ 0, & i \notin I_j. \end{cases} \quad (21)$$

3. Update basis functions

$$P_{ij}^{n+1} = P_{ij}^n + d_{ij} \quad (22)$$

4. Define local error outside of boundary regions,

$$e_j = \max_i(|\hat{d}_{ij}|), \quad i \notin G \quad (23)$$

5. If  $\|\mathbf{e}\|_\infty > \text{tol}$ , go to Step 1

In practice, checking for convergence should only be done every tenth iterations or so, since a single iteration has negligible cost.

Because we have explicitly enforced partition of unity in cells belonging to the global boundary, we must now show that the updates  $\hat{d}_{ij}$  preserve the same property in cells not on the boundary. We will assume that the row sum of the matrix used for the iterations is zero and that the initial prolongation operator has partition of unity,

$$\sum_j A_{ij} = 0, \quad \sum_j P_{ij}^0 = 1 \quad \forall i. \quad (24)$$

If needed, one can easily ensure that the iteration matrix has zero row sum by adjusting the diagonal elements. This requirement also applies to the classical MsFV method, so the same adjustment is used if special basis functions for wells, boundary conditions, and compressibility are not constructed. We can then write out the explicit update for a single cell  $i$ , summed over all coarse blocks,

$$\begin{aligned} \sum_j P_{ij}^{n+1} &= \sum_j P_{ij}^n - \frac{\omega}{A_{ii}} \sum_j \sum_k A_{ik} P_{kj}^n \\ &= 1 - \frac{\omega}{A_{ii}} \sum_k A_{ik} \left( \sum_j P_{kj}^n \right) = 1 - \frac{\omega}{A_{ii}} \sum_k A_{ik} = 1, \end{aligned}$$

showing that the update always preserves partition of unity.

**Iterative multiscale.** Multiscale finite-volume methods have a link to multigrid methods in the sense that they can be used as two-level methods in combination with a *smoother* step that takes care of localized errors. This can be used for error control, to treat compressibility and nonlinearity, or by using the multiscale solver as a pure linear solver for the fine-scale system. Herein, we will only use a smoother to handle sub-grid effects, such as improving the approximation of local features in the pressure field or to reduce prolongation errors. While it is fully possible to use the MsRSB prolongation operator as a linear solver for the fine-scale system, see (Møyner 2014), we will herein only use the multiscale solver as an approximate solver.

To define an iterative scheme, we let the solution at step  $n$  be denoted  $\mathbf{x}^n$  and introduce the defect

$$\mathbf{d}^n = \mathbf{b} - A\mathbf{x}^n, \quad (25)$$

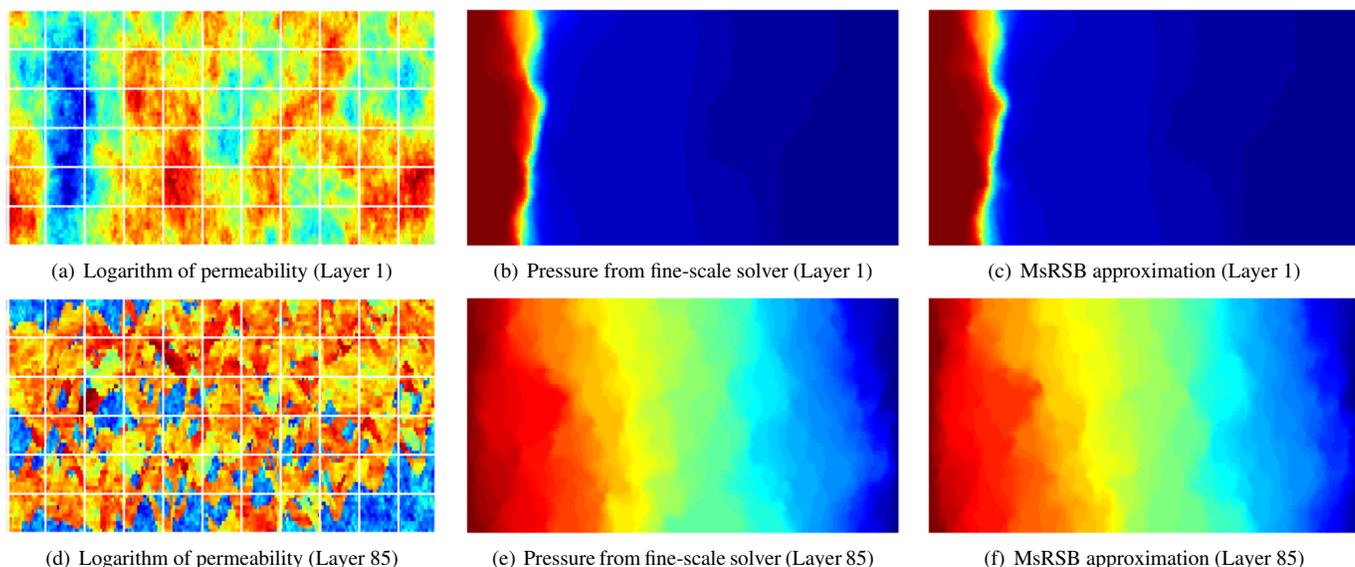
If we let  $\mathbf{y}^n = S(\mathbf{d}^n)$  denote the smoother applied to the defect with initial guess zero, we can then write the next update as the previous solution with the smoothed update added in, along with a *coarse correction* that ensures that the update does not remove the coarse-scale conservative property of the solution,

$$\mathbf{x}^{n+1} = \mathbf{x}^n + P(A_c^{-1}R(\mathbf{d}^n - A\mathbf{y}^n)) + \mathbf{y}^n. \quad (26)$$

This iterative scheme relies on an inexpensive smoother for the updates. We use a single pass of incomplete LU-factorization with zero fill in (ILU0) for the multiphase problems and five Jacobi iterations for the single-phase problems. Such a single smoother pass plus the coarse correction will be termed a multiscale cycle.

## Numerical Experiments

The MsRSB method introduced above has been implemented using the Matlab Reservoir Simulation Toolbox (MRST), see (Lie et al. 2012; Krogstad et al. 2015), and will be released as free, open-source software in 2015. In the following we will report the results of a series of numerical experiments we have run to validate the multiscale formulation, verify our implementation, and demonstrate the utility of the resulting solver. Our test matrix includes flow physics ranging from incompressible, single-phase flow to a three-phase, compressible gas injection and consider both simplified Cartesian grids and corner-point geometries of real oil fields.



**Figure 4—Comparison of pressure solutions computed by the fine-scale solver and the MsRSB method on the top and bottom layer of the second SPE 10 data set. Flow is driven by a difference in the fixed pressures specified at the left and right boundaries.**

**Accuracy for single-phase flow.** To assess the spatial accuracy of the MsRSB method, we consider the simple, single-phase flow model in Eq. 1 applied to two different test cases: Model 2 from the 10th SPE Comparative Solution Project (Christie and Blunt 2001) was originally designed as a challenging benchmark for upscaling methods and is today the de facto benchmark for new multiscale methods. The model is described on a uniform Cartesian grid, but has highly heterogeneous petrophysical properties sampled from a Brent sequence. The second test case uses the grid geometry and petrophysical properties from a simulation model of the Gullfaks Field, which contains the same type of Brent sands, but is structurally very complex. For both models, we will measure the discrepancy between the multiscale approximation and the fine-scale reference solution using scaled  $L_\infty$  and  $L_2$  norms,

$$\|p^{fs} - p^{ms}\|_\infty = \frac{\max_{i \in F} |p_i^{fs} - p_i^{ms}|}{\max_{i \in F} |p_i^{fs}|}, \quad \|p^{fs} - p^{ms}\|_2 = \sqrt{\frac{\sum_{i \in F} |p_i^{fs} - p_i^{ms}|^2 |\Omega_i|}{\sum_{i \in F} |p_i^{fs}|^2 |\Omega_i|}}, \quad (27)$$

where  $p_i^{fs}$  and  $p_i^{ms}$  denote the pressure values computed in cell  $\Omega_i$  by the fine-scale and the multiscale methods, respectively. Discrepancies in fluxes are defined analogously.

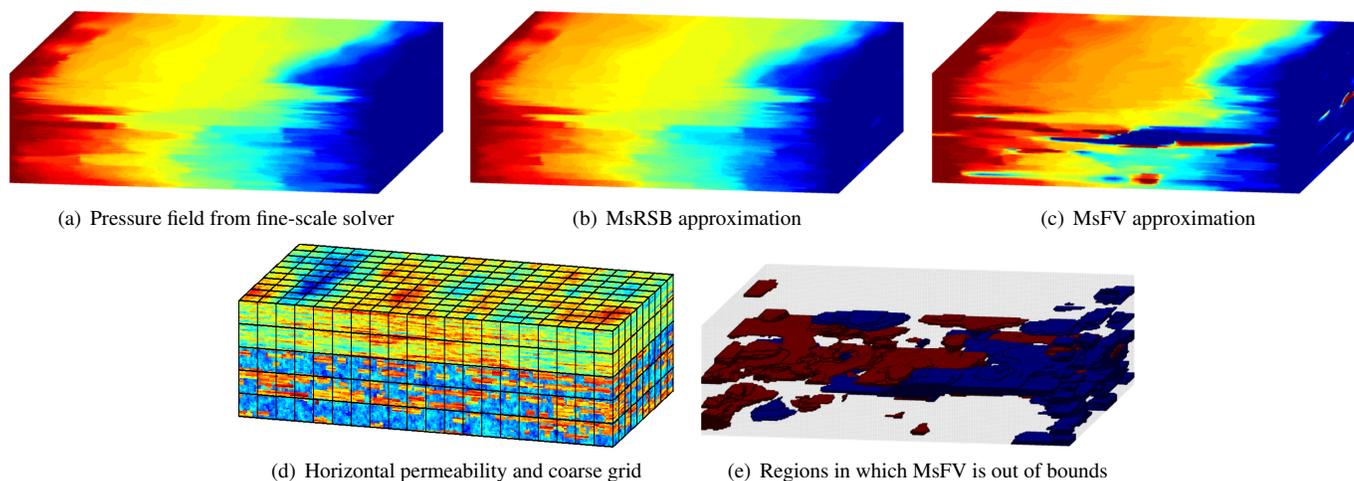
**Second SPE 10 data set:** The synthetic reservoir is described by a  $60 \times 220 \times 85$  Cartesian grid with cells of uniform size  $20 \times 10 \times 2$  ft<sup>3</sup>. The reservoir contains two sands with very different heterogeneity. In the Tarbert formation found in the top 35 layers, the permeability follows a lognormal distribution, giving smoothly varying heterogeneities that are resolved quite well by most multiscale methods. The Upper Ness formation found in the bottom 50 layers is fluvial and consists of an intertwined pattern of long and high-permeable sand channels interbedded with low-permeable mudstone. The combination of very long correlation lengths and many orders-of-magnitude difference in permeabilities between neighboring cells makes Upper Ness very challenging to resolve accurately.

First, we consider a two horizontal  $60 \times 220$  slices with isotropic permeabilities sampled from the top and bottom layers of the model. We apply a fixed pressure at the left and right boundaries to drive flow across the domain. The domain is partitioned into coarse blocks made up of  $10 \times 20$  fine cells so that the coarse grid blocks are square in size. Because we do not use correction functions, we also add a thin layer of coarse blocks near the edges of the domain where the boundary conditions are imposed to ensure that the basis functions do not get regions of constant value near the edges. This gives a total of 78 coarse blocks, which corresponds to an upscaling factor of 170.

**Fig. 4** shows the permeability for both layers and compares the pressure fields computed by the fine-scale solver and MsRSB using a single multiscale solve without subsequent iteration cycles. **Table 1** reports the corresponding discrepancies measured in the relative  $L_2$  and  $L_\infty$  norms defined in Eq. 27. For comparison, we also report discrepancies for the original MsFV method as implemented in the `msfv` module of MRST, see (Møyner and Lie 2014a) for details. Whereas the solution quality is generally very good for both solvers for Tarbert, MsRSB clearly outperforms MsFV on Upper Ness. Several authors have independently shown that the MsFV method has issues with coarse scale stability in the presence of channelized, high-contrast formations and will suffer from strong unphysical oscillations that may prevent iterative versions of the method from converging properly, see

**Table 1—Discrepancy between the fine-scale solution and approximate solutions computed by the MsFV and the MsRSB methods for a 2D test problem with permeability sampled from the top and bottom layer of the second SPE 10 data set; see Fig. 4.**

Setup of simulation		Pressure		Flux	
Model	Solver	$L_2$	$L_\infty$	$L_2$	$L_\infty$
Tarbert (Layer 1)	MsFV	0.026	0.202	0.210	0.597
Tarbert (Layer 1)	MsRSB	0.023	0.197	0.165	0.508
Upper Ness (Layer 85)	MsFV	0.234	1.511	1.056	1.940
Upper Ness (Layer 85)	MsRSB	0.039	0.075	0.296	0.427



**Figure 5—Comparison of pressure solutions computed by the fine-scale solver and the MsFV and MsRSB methods on the full SPE 10 model. Flow is driven by a difference in fixed pressures specified at the west and east boundaries.**

e.g., (Sandvin et al. 2013; Wang et al. 2014; Møyner and Lie 2014a) and references therein. MsRSB is much more robust and does not suffer from such problems and therefore gives solutions of similar accuracy for the smooth and channelized layers.

For the full 3D model the flow patterns become more complex, partially because of strong anisotropy, which together with higher aspect ratios pose additional challenges for multiscale methods. To coarsen the 1.1 million cells in the fine-scale model, we use the same strategy as in 2D with  $10 \times 20 \times 5$  fine cells per coarse block, giving coarse blocks of size  $200 \times 200 \times 10$  feet. For the boundary conditions we will use the same values as in the previous example.

The results are in line with what we observed in 2D: Both MsFV and MsRSB perform reasonably well on the upper part of the model, but the MsFV method becomes unstable in the channelized formation at the bottom of the model. Looking at Fig. 5 it is difficult to distinguish qualitative differences between the MsRSB and the reference solution, which is also confirmed by the quantitative comparison in Table 2. SPE 10 is a challenging benchmark in terms of heterogeneity, and the good accuracy obtained with the MsRSB method without any kind of grid adaption or smoothing iterations is remarkable.

Møyner (2014) reports a more thorough comparison of the MsRSB, MsFV, and MsTPFA prolongation operators for various 2D and 3D subsets of the SPE 10 model, using pure multiscale solvers or as part of a GMRES iterative solver. The results show that the MsFV method gives accurate multiscale solutions and converges rapidly on the smooth Tarbert formation, and if the original mass-conservative control-volume restriction operator used to define the coarse-scale system is replaced by a Galerkin restriction operator, the method is accurate and efficient also on Upper Ness. The MsTPFA method is more robust but less accurate and efficient than MsFV on Cartesian partitions. However, by introducing a coarse partition that adapts to contrasts in the fine-scale transmissibilities, the MsTPFA method becomes as accurate and efficient as MsFV (which generally cannot be applied to such adapted grids). In all tests, however, the MsRSB method performs equally well or better than the other two

**Table 2—Discrepancy between the fine-scale solution and approximate solutions computed by the MsFV and the MsRSB methods for the full SPE 10 model, see Fig. 5.**

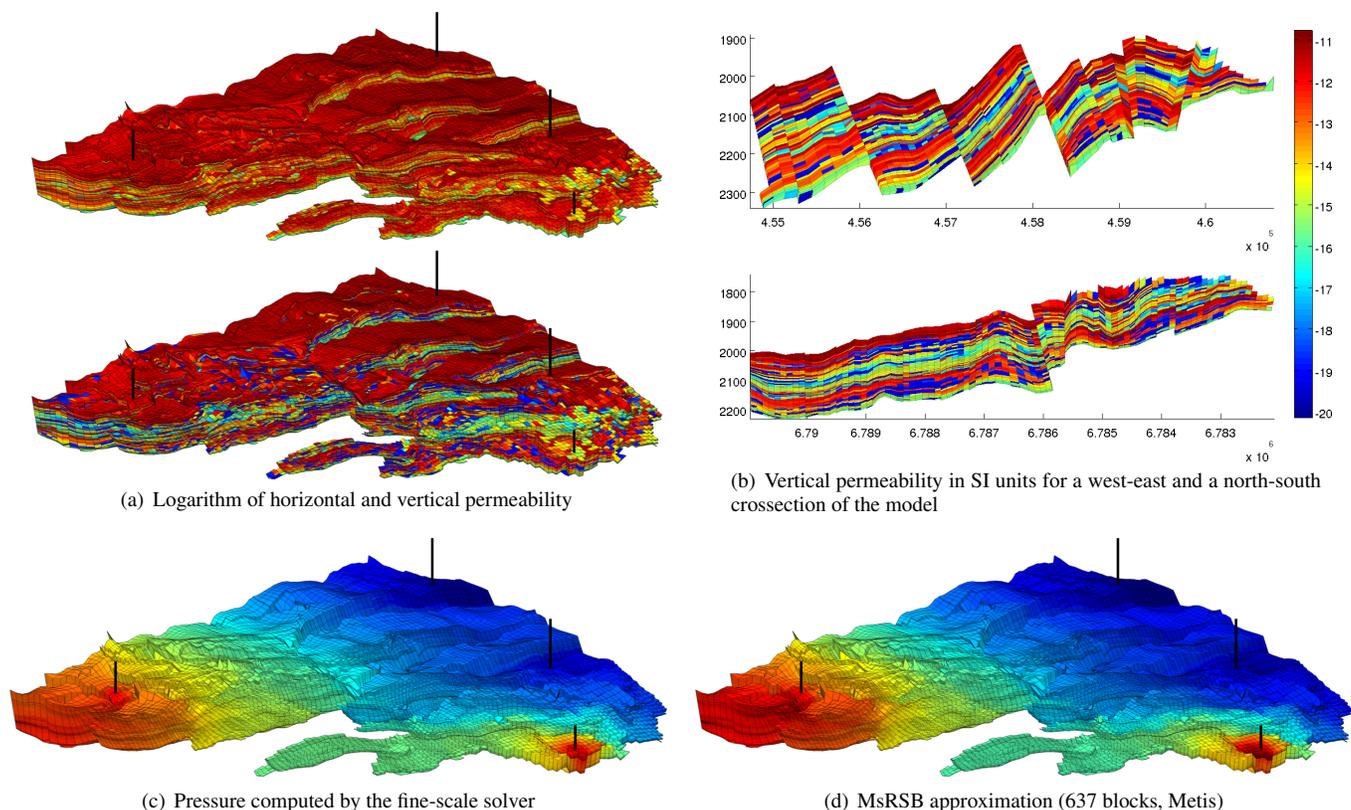
Solver	Pressure		Flux	
	$L_2$	$L_\infty$	$L_2$	$L_\infty$
MsRSB	0.042	0.361	0.641	1.060
MsFV	2.470	171.8	2.627	7.270

methods and is less affected by the choice of restriction operator and type of coarse partition.

**Gullfaks field model:** The Gullfaks Field is an oil and gas field located in the Norwegian sector of the North Sea that produces primarily from Brent sands, i.e., the same type of sedimentary environments as seen in the SPE 10 model. However, unlike SPE 10, the Gullfaks Field is structurally very complex and contains a large number of sloping faults, with angles varying from 30 to 80 degrees, and throws from zero and up to three hundred meters. The simulation model is represented on a  $80 \times 100 \times 52$  corner-point grid in which 216 334 cells are active. Almost 44% of the cells have non-neighboring connections, and when the corner-point grid is turned into a matching polyhedral grid, the number of cell faces range from four to thirty-one. The combination of strong heterogeneity, large anisotropy and aspect ratios, degenerate cell geometries, and unstructured grid topology makes the Gullfaks model very challenging for any multiscale solver. The flow is driven by four wells, two producers and two injectors, that are placed quite arbitrarily near the model perimeter to create significant pressure drops across the majority of the field. The wells are perforated through all layers of the model.

We consider three different partitions. The first is a coarse grid initially partitioned with  $15 \times 15 \times 20$  fine cells per coarse block. The large coarsening factor in the vertical direction is chosen because of the many inactive layers and cells. Any coarse block intersected by a fault is then split into two, resulting in a semi-structured coarse grid with 659 blocks. The second coarse grid is constructed by the use of Metis (Karypis and Kumar 1998) to partition the transmissibility graph with a target set to 659 blocks to match the number of blocks in the structured partition. We also consider another Metis partition with approximately four times as many coarse blocks (2192 after processing) to demonstrate how the accuracy of the MsRSB method can be improved by coarse mesh refinement. We emphasize that no manual effort was required to create the three coarse partitions.

The approximate solution computed on the coarsest Metis-based grid can be seen in **Fig. 6**, while **Table 3** reports the discrepancy between the fine-scale solution and the approximate multiscale solutions computed on the three different coarse grids. We note that the multiscale approximations are quite accurate and that refining the coarse mesh improves the accuracy at the cost of a larger coarse system. The table also reports discrepancies after applying five multiscale cycles of 10 Jacobi iterations each, which show that local errors are quickly removed by the inexpensive smoother. Altogether, the results are very promising in view of the combined challenge posed by the extremely complex and partially degenerate geometry and the large permeability contrasts. Being able to handle models of this level of structural and stratigraphic complexity in a robust and automated fashion is essential if the goal is to bring multiscale methods closer to practical usage. The interested reader can also consult (Møyner 2014) for a discussion of MsRSB used as an iterative linear solver for a case with seven injection and eleven production wells.



**Figure 6—Computation of incompressible, single-phase pressure distribution using the grid geometry and the petrophysical data from a simulation model of the Gullfaks field.**

**Table 3—Discrepancy between the fine-scale solution and approximate solutions computed by the MsRSB method on the Gullfaks case; see Fig. 6.**

Solver	Coarse grid		Initial pressure		Pressure, 5 cycles	
	Block type	DoF	$L_2$	$L_\infty$	$L_2$	$L_\infty$
MsRSB	$15 \times 15 \times 20$	659	0.052	4.166	0.048	0.170
MsRSB	Metis	659	0.018	0.111	0.017	0.050
MsRSB	Metis	2192	0.012	1.188	0.008	0.009

**Numerical results for multiphase flow.** Solving for the pressure is straightforward for the single-phase problem, but if we want to use the multiscale operators for nonlinear multiphase flow we need to treat the inherent nonlinearity. We employ the most straightforward approach and simply reduce the fine-scale Jacobian in the same manner as for single-phase flow, i.e.,  $J_c = RJ_pP$ . The update to the pressure is computed using one or more multiscale cycles, either until we have reached the maximum allowed number of cycles or the linear solution has reached some convergence criterion. In each example, we will point out which of the two alternatives we use.

To determine when the nonlinear loop has converged, the correct approach would be to apply the following fine-scale condition,

$$\|\mathcal{R}_p(p^n)\|_\infty \leq \text{tol}_p. \quad (28)$$

However, when solving the multiscale problem it is more natural to look at the residual on the coarse scale instead,

$$\|\mathcal{R}\mathcal{R}_p(p^n)\|_\infty \leq \text{tol}_p. \quad (29)$$

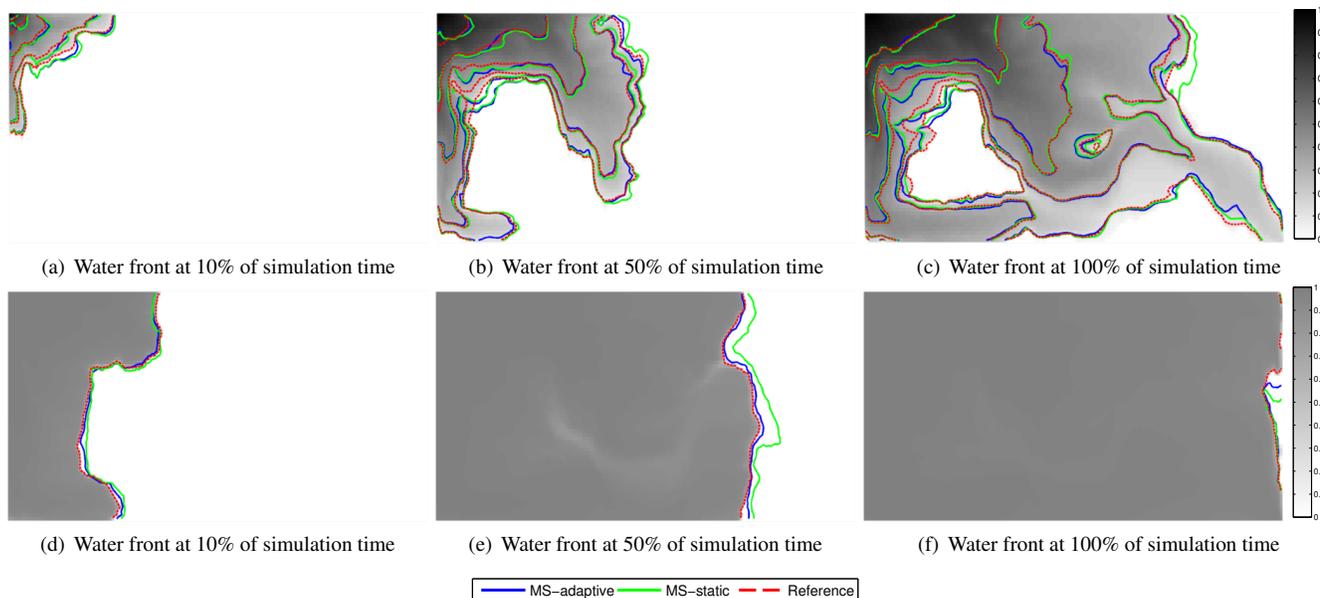
Here, local errors can sum up inside each control volume, making the convergence criterion similar to what an upscaled model would have used. The difference is that we in the multiscale method have approximate pressures on the fine grid that can be used for the transport step. As in the single-phase case, the pressure field produced by the multiscale cycles is generally not an exact solution, and after the convergence of each nonlinear step we therefore need to solve a set of local Neumann problems to reconstruct a conservative velocity field that can be passed onto the transport solver. In this reconstructive step, we evaluate pressure-dependent properties using the prolonged pressure, thereby making the problem linear and simpler to solve.

**SPE 10 – water flooding:** In our first multiphase example, we return to the top layer of the SPE 10 model. The 2D reservoir is initially filled with pure oil and is produced by a quarter five-spot well pattern with water injection in one corner and fluid production in the diagonally opposite corner. The wells are controlled by pressure, which gives a more challenging test than rate-based injection since the inflow and outflow now depends directly on the pressure distribution. Basis functions can be constructed for almost arbitrarily complex partition and to improve our prediction of the pressure distribution, we adapt the coarse grid by applying a radial refinement near the wells, but keep the structured coarse grid from the single-phase example elsewhere as shown in **Fig. 8(a)**.

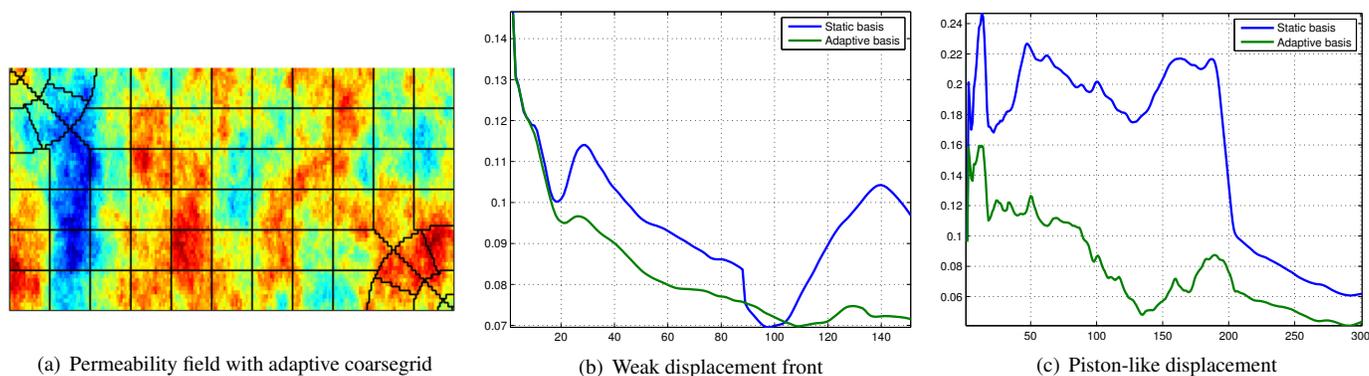
There are two different ways of using a multiscale solver for time-dependent problems. We can either keep the same basis functions throughout the whole simulation, or update them to account for dynamic changes in mobility. The usual way to do this is to recompute basis functions locally whenever the saturation changes significantly. Smoothed basis functions, however, do not need to be completely regenerated: Because the underlying iterative process can start from any function having partition of unity, we can simply restart the iteration process with changed mobilities and continue until the basis functions are sufficiently smooth again. To compare the two approaches, we consider two fluid models that exhibit drastically different behavior. In the first case, the injected fluid has one tenth the viscosity of the reservoir fluid, resulting in viscous fingering and a weak displacement front, but relatively smooth changes in the mobility inside each coarse block. The second case is the complete opposite with an injected fluid that is ten times more viscous than resident fluid. This will result in a piston-like displacement that induces large changes in the total mobility as the strong leading shock propagates through a coarse block. As noted by Kippe et al. (2008), the multiscale method must therefore accurately account for dynamic changes in the mobility to be able to correctly predict the speed of the displacement front.

**Fig. 7** shows the saturation of the injected fluid for both mobility ratios at three different intervals. For the weak displacement front there is good agreement between the multiscale and the reference solutions, even if basis functions are not updated throughout the simulation. For the sharp front, however, we can see that the multiscale method with static basis functions overestimates the speed of the leading shock. This error accumulates during the simulation and results in earlier breakthrough. From the plot of the saturation error over time in **Fig. 8**, we see that the error is larger with static basis functions in both cases, but the difference in errors is more pronounced for the piston-like displacement.

**Norne field model – water flooding:** In the next example, we consider a corner-point model that represents the Norne Field in the Norwegian Sea, see (Norne 2012; Package 2). The model contains multiple faults and pinched cells, and the top layers are almost completely separated from the layers below. The layered permeability distribution varies largely along the vertical direction giving significant differences in fluid velocities.



**Figure 7—Saturation profiles for the two-dimensional waterflood example. The reference saturation field is shown in graytones, with contour lines of the the two multiscale solutions superimposed. The top and bottom rows correspond to weak and piston-like displacement respectively.**



**Figure 8—Coarse grid and errors in  $s_w \phi$  over time for water injection in the top layer of the SPE 10 model.**

We will neither use the fluid description nor the well positions from the real simulation model, but instead consider a feigned water flooding in which we have set the time period to one hundred years to ensure that steady state flow is achieved at the end. We have two injectors operating at fixed bottom-hole pressure of 375 bar and a producer in the opposite end operating at 250 bar. The fluid model is a two-phase system with quadratic relative permeabilities, gravity, and significant oil compressibility so that the oil density increases by 30% from the minimum to maximum pressure observed in the reservoir.

We will consider three different multiscale solvers. The first uses a fixed tolerance for the iterative convergence, set to a relatively loose value of 0.1. This means that for each linear solution we keep iterating until

$$\frac{\|A\mathbf{x}^n - b\|_\infty}{\|b\|_\infty} \leq 0.1. \quad (30)$$

The other solvers use a fixed number of iterations for each linear solve, with one and five cycles, respectively. In either case the solver will be far from converged in any conventional sense, but the goal is to capture the main flow features at a reduced computational cost, not to converge to the exact solution.

To validate the three multiscale solvers, we will examine the predicted injection and production curves, which in many workflows are the output that matter most to the end user. Notice, however, that well rates and bottom-hole pressures can be very sensitive to the mobility in the well cells, values that are likely not exact even though the general reservoir state is accurately reproduced by the multiscale solver. Various well curves are shown in **Fig. 9**. For scale, we have also included well curves predicted by fine-scale, sequential and full-implicit simulations. Because the wells are controlled by bottom-hole pressure, we

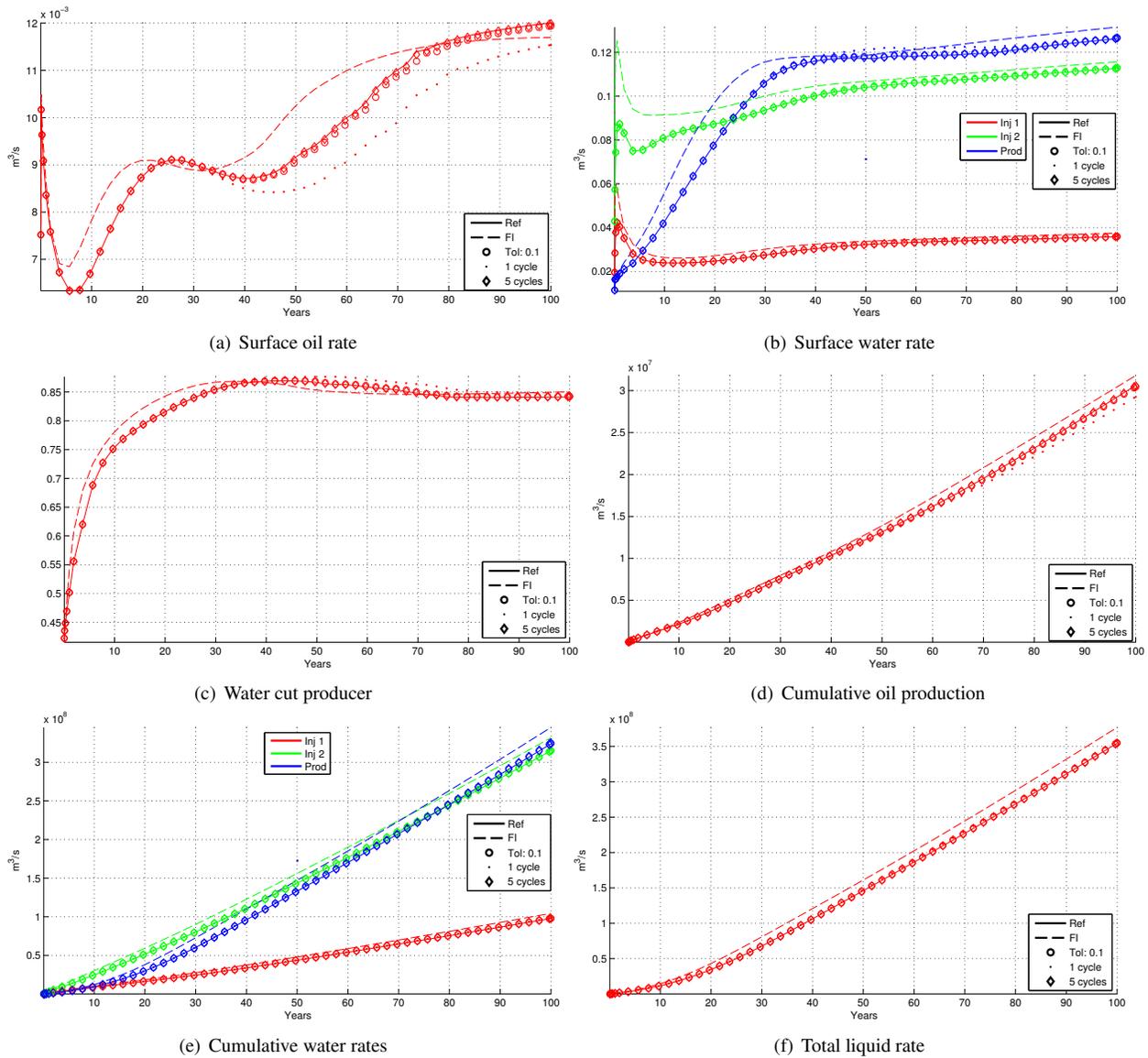


Figure 9—Well curves for water flooding for a modified model of the Norne field. Markers indicate multiscale solutions whereas the solid and dashed lines are fine-scale solution computed by a sequential and a fully-implicit simulators.

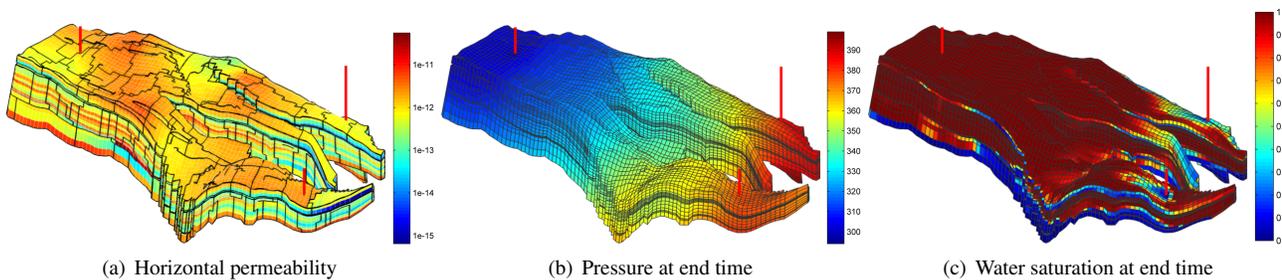


Figure 10—Water flooding for a modified model of the Norne field.

**Table 4—Timing results for the Norne simulation example. We compare both the time spent in the linear solver and the total solution time. For the multiscale solver, the solve time also includes the costs of computing basis functions and solving the linear system for the reconstruction of mass-conservative, fine-scale fluxes. The latter cost is also given specifically in parenthesis.**

Solver	Time spent (minutes)				Speedup, solve time		Speedup, total time	
	pressure	transp.	other	total	seq.	implicit	seq.	implicit
Implicit w/AMG-CPR		19.24	3.51	22.75	—	—	—	—
Sequential w/AMG	2.56	0.07	2.32	4.95	—	7.32	—	4.60
MS, tol 0.1	0.49 (0.26)	0.07	2.84	3.40	4.68	34.22	1.45	6.68
Ms, 1 cycle	0.36 (0.25)	0.08	2.61	3.04	6.10	44.64	1.63	7.47
Ms, 5 cycles	0.44 (0.25)	0.07	2.53	3.05	5.16	37.72	1.62	7.47

plot both well rates and derived quantities such as water cut and cumulative production for both oil and water. We can see that the multiscale approximations are quite close to the sequential reference solution and that the largest difference is between the two different fine-scale solutions. There are minor differences in the oil rate predicted by the single-cycle multiscale solver, but otherwise it is hard to distinguish curves in any meaningful way. In other words, to obtain a better match with the fully implicit solver, one should work with the splitting and the time-stepping strategy rather than the multiscale spatial discretization.

Before we compare the computational efficiency of the five solvers, we mention a few caveats: The sequential and fully-implicit solvers are both written in Matlab using automatic differentiation (Krogstad et al. 2015), so we expect there to be considerable overhead in evaluating fluid properties, assembling Jacobians dynamically, and so on. These parts of the solvers are not written for speed, but to provide great flexibility in changing fluid models and (nonlinear) solution strategies. Secondly, the linear solver used in the fine-scale simulation is an algebraic multigrid method written in Fortran (Notay 2010) using default settings, whereas the multiscale solver is written in Matlab. However, the basis functions are computed using a straightforward C-accelerated implementation. The sequential transport solver uses GMRES preconditioned with ILU(0). With these factors in mind, we can look at **Table 4** and compare runtimes.

Looking only at the speedup for the solvers, we observe that the sequential method is seven times faster than the fully-implicit method, which uses the same algebraic multigrid method in combination with a CPR preconditioner. Comparing the sequential multigrid solves and the multiscale solver, we see that the latter is five to six times faster. In total, this means that the multiscale solver with the best match is a factor thirty to forty times faster than the fully-implicit method when looking only at the time for the linear solver. In all simulations, a large amount of time is spent on property evaluation and assembly of linearized equations. Looking at the total simulation time, the sequential method is therefore four to five times faster than the fully-implicit method, whereas the multiscale method is only one and a half times faster than the sequential method and six to seven times faster than the fully-implicit method. However, the time spent outside linear solvers is roughly the same in all sequential simulations and this large constant term can be significantly reduced if one is willing to manually derive and code the Jacobians of the linearized equations and offload property evaluations to compiled code.

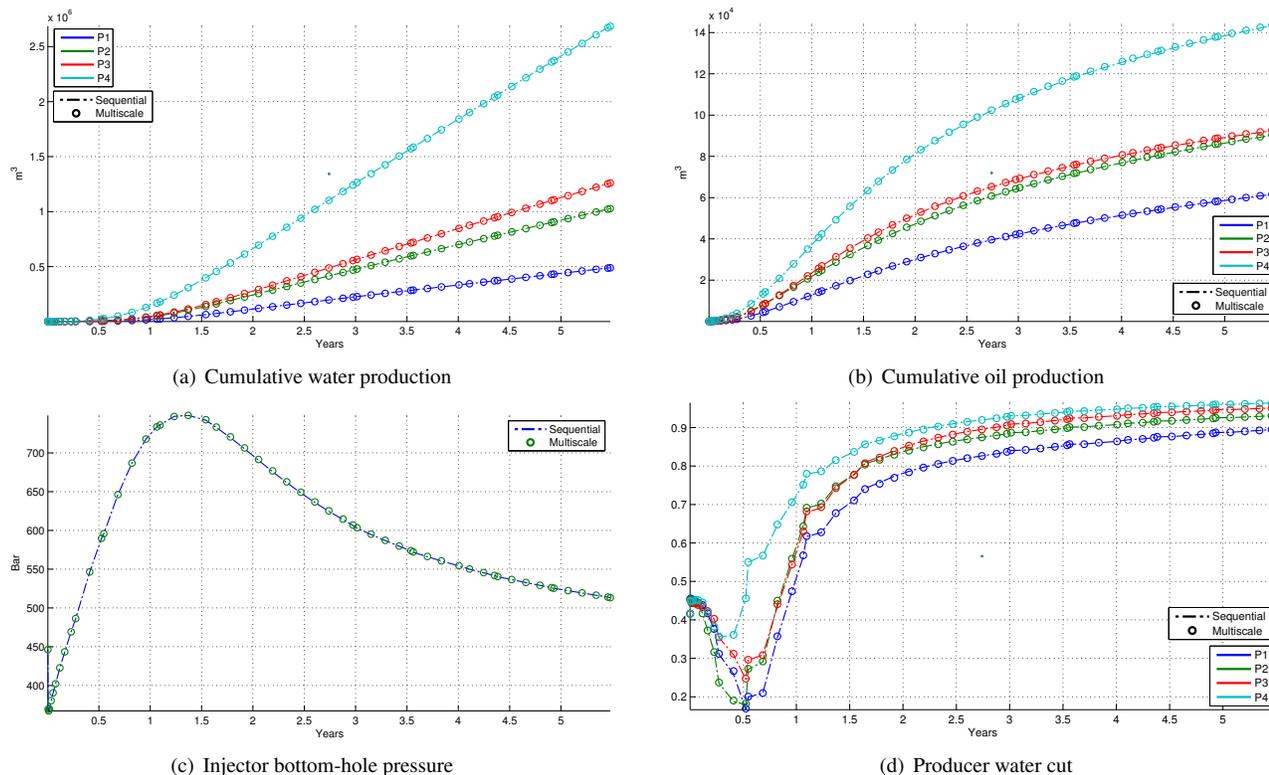
**SPE 10 – water flooding, revisited:** The Norne model is relatively small and hence the timing results are not dominated by the linear solver. In addition, we therefore report a timing experiment on the full SPE 10 model. To enable us to test what is essentially a similar case to the Norne model, but with more heterogeneity and a larger number of cells, we use the same fluid model as in the Norne study, which is more compressible than in the original SPE 10 benchmark. We only compare the sequential solver to the multiscale solver with a fixed tolerance. Because of the large contrasts in the model, we reduced the tolerance in the AMG solver from  $10^{-6}$  to  $10^{-5}$  to get reasonable runtime for the reference solution.

**Fig. 9** shows that there is almost no discrepancy between the production and injection curves predicted by the sequential fine-scale and the multiscale simulators. On the other hand, **Table 5** shows that the fine-scale simulation takes very long time, which primarily can be attributed to the pressure solver. The Jacobians produced for SPE 10 can be very challenging to solve accurately, which is reflected in the runtime. The multiscale solver is much faster since it does not need to reduce the residual to a low tolerance before updating saturations. The time spent in the transport solver is comparable for both simulations, indicating that the multiscale solver produces high quality velocity fields. In terms of speedup, the multiscale pressure solver is nineteen times faster than fine-scale algebraic multigrid, whereas the total speedup is approximately seven times for the entire simulation. The computational cost categorized as 'other' in the table is almost twice as big for the multiscale solver than for the fine-scale solver. Our implementation uses certain sparse matrix operations to construct the local problems for the flux reconstruction that have acceptable overhead for small models, but become very costly for models with a million cells or more. These operations, however, are both almost perfectly parallel and can easily be avoided when the methods were implemented in a compiled language.

**Modified SPE 1 – gas injection:** In the final case we consider gas injection using fluid properties taken from the SPE 1 benchmark (Odeh 1981), which is a black-oil model with nonlinear relative permeabilities, pressure-dependent viscosity, and gas dissolution. As the original SPE1 problem has homogeneous permeability field and consists of only  $10 \times 10 \times 3$  without any

**Table 5—Timing results for 2000 days of water injection on the full SPE 10 model with the same fluid model as the Norne case, which is more compressible than the fluid model of the original benchmark problem. The number in parenthesis is the time required to reconstruct mass-conservative, fine-scale fluxes.**

Solver	Time spent				Speedup solve time	Speedup total time
	pressure	transport	other	total		
Sequential w/AMG	18 h 19 min	25 min	1 h 8 min	19 h 53 min	-	-
MS, tol 0.1	34 min (7 min)	24 min	1 h 56 min	2 h 55 min	19.2	6.8



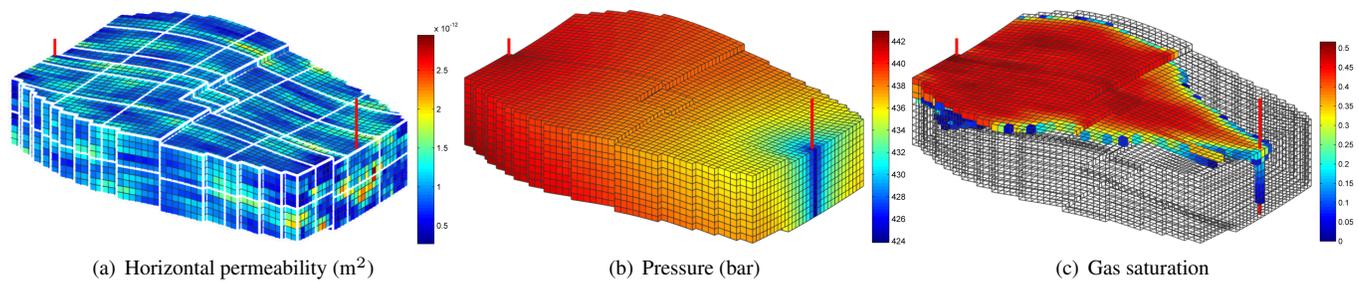
**Figure 11—The well production curves for the modified SPE10 benchmark using the same fluid model as in the Norne case. A single multiscale solution is compared to the sequential simulation.**

structural complexity, we replace the grid by a synthetic corner-point model that contains four intersecting faults with modest throws and a lognormal permeability distribution, as shown in **Fig. 12**. The permeability in the vertical direction is halved to make the problem slightly anisotropic and the bounding box of the grid is  $3000 \times 3000 \times 30$  meter. In logical indices the grid is discretized into  $50 \times 50 \times 5$  equally sized cells, resulting in just over 20,000 active cells in simulation model.

We place an injector to the far west and a producer at the far east. The injector will inject gas for five years into the under-saturated and mostly oil filled reservoir with a fixed rate corresponding to 2% of the pore volume per day at surface conditions, which is about  $4000 \text{ m}^3$  per day at reservoir conditions. The producer operates at a fixed bottom-hole pressure.

For the multiscale solver we select a simple partition consisting of  $10 \times 10 \times 5$  fine cells per coarse block, as seen in **Fig. 12(a)**. Our primary concern here is to test the solver on a model with realistic flow physics, including significant compressibility, gravity, and gas dissolution. We run the same simulation multiple times with different tolerances. On the low end, we have the tolerance of 0.1 that gave acceptable results for the moderately compressible oil/water model on Norne. We know that this will only result in a few iterations per time step and will probably not be sufficient to correctly resolve the dynamics of the gas front, which is sensitive to the accuracy of our pressure approximation because both the compressibility and the dissolution model have significant pressure dependencies. For instance, if the pressure is higher than the reference value, some of the injected gas will remain in the oil phase, resulting in differences in mobility. We then gradually reduce the tolerance to 0.05, 0.01, and 0.001. Setting the relative tolerance to 0.001 is fairly close to pure linear solver territory, but the other tolerances should be considered approximate solutions and hence we do not expect a perfect match.

Looking at the well curves in **Fig. 13** we see that the multiscale solvers qualitatively produces the correct graphs. However, the lower tolerances (0.1 and 0.05) predict too late breakthrough of the gas, most likely because of a somewhat over-estimated pressure seen at the injector. The pressure buildup results in less gas flowing freely, further increasing the pressure buildup during



**Figure 12—A modified version of the SPE 1 benchmark posed on a faulted reservoir geometry. The vertical dimension of the grids has been exaggerated by a factor twenty to improve visualization.**

the simulation. With tolerance 0.001 the multiscale solver is very close to the reference solution, whereas the 0.01 case being somewhere in between. On average, the lowest tolerance used approximately nine iterations per linear solve, while the strictest tolerance used seventy. This number could likely be reduced by the use of GMRES for acceleration. Generally, we see that the multiscale solver is able to solve the relatively complex fluid physics of the problem without any convergence issues even if the pressure solution is only approximate. However, because of the problem is strongly and nonlinearly pressure dependent, more iterations are required than for the Norne case.

### Concluding Remarks

We have presented a novel multiscale formulation that uses an iterative process to construct basis functions. The iterative process is defined using standard matrix manipulations in combination with simple grid indicators that can easily be obtained from general coarse partitions without any stringent requirements on the underlying topology. Because the formulation requires only knowledge of cell centroids and topological neighborhood, the method is applicable to general polyhedral grids, with no restrictions when it comes to handling erosion, pinched cells, or faults.

Extensive numerical tests on single-phase problems, a few are reported herein and in (Møyner 2014), show that the MsRSB method compares favorably with the classical MsFV in terms of accuracy, but is generally much more robust and less affected by the choice of restriction operator and the resolution and type of the coarse partition. Notably, the pressure field for the challenging SPE 10 benchmark is solved to within 5% accuracy without the need for additional iterations or coarse grid adaption. Likewise, a high degree of accuracy is observed on the very structurally complex Gullfaks field model, for which a wide range of coarse partitions can easily be generated using standard graph partitioning algorithms, which indicates that extensive manual tuning may not be necessary for complex models.

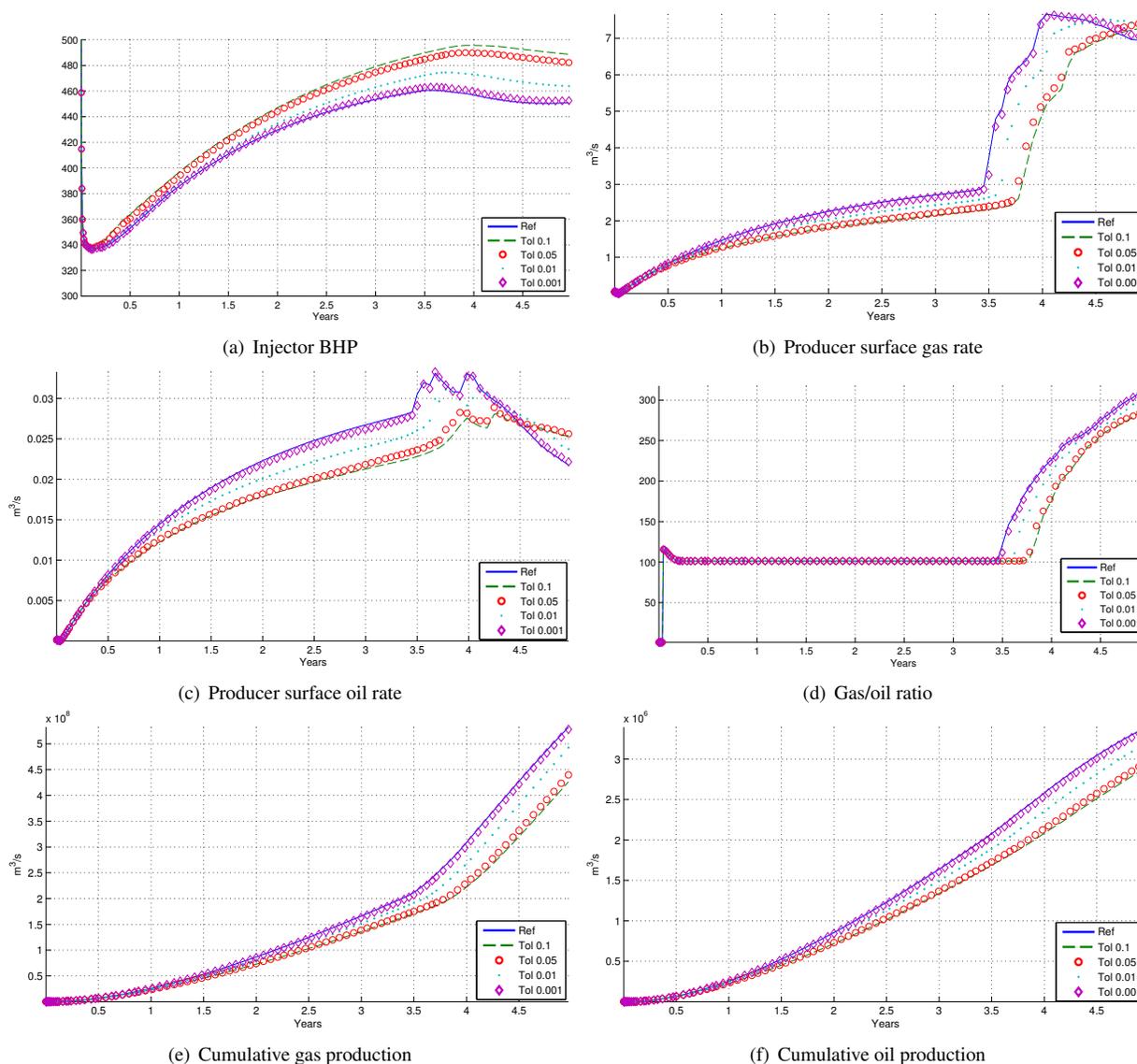
Through a set of multiphase test cases we have also demonstrated that the method can easily adapt to wells, changing mobilities and complex fluid behavior such as strong compressibility, gravity, and gas dissolution. The multiscale solutions generally agree well with the fine-scale reference solution, and by adjusting the multiscale tolerance, a perfect match can be observed at the cost of more iterations. The computational efficiency of the method compares favorably to an algebraic multigrid solver, mainly because accurate fine-scale fluxes can be computed regardless of whether the pressure has converged to a low residual or not. However, we note that the multiscale solutions will generally be different from fine-scale solutions obtained by a fully-implicit simulator. However, these differences are attributed to the sequential formulation and we therefore have some confidence that the multiscale solution can efficiently approximate complex physics problems posed in a sequential form. If the goal is convergence to the same production curves as a fully-implicit simulator, the issue lies with the sequential formulation rather than the multiscale component.

### Acknowledgments

This research was funded in part by Schlumberger Information Solutions and by the Research Council of Norway under grant no. 226035. We thank Statoil (operator of the Norne field) and its license partners ENI and Petoro for the release of the Norne data and acknowledge the Center for Integrated Operations at NTNU for cooperation and coordination of the Norne cases. We also thank Jostein R. Natvig and Halvor Møll Nilsen for constructive discussions.

### References

- Aarnes, J. E. 2004. On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation. *Multiscale Model. Simul.*, 2(3):421–439. doi: 10.1137/030600655.
- 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.
- Alpak, F. O., Pal, M., and Lie, K.-A. 2012. A multiscale method for modeling flow in stratigraphically complex reservoirs. *SPE J.*, 17(4):1056–1070. doi: 10.2118/140403-PA.



**Figure 13—Well curves for the modified SPE 1 test cases for the fine-scale sequential solver (termed ref) as well as multiscale solvers with different convergence criterions.**

Arbogast, T. 2002. Implementation of a locally conservative numerical subgrid upscaling scheme for two-phase Darcy flow. *Comput. Geosci.*, 6(3-4):453–481. doi: 10.1023/A:1021295215383.

Arbogast, T., Pencheva, G., Wheeler, M. F., and Yotov, I. 2007. A multiscale mortar mixed finite element method. *Multiscale Model. Simul.*, 6(1):319–346. doi: 10.1137/060662587.

Babuka, I., Caloz, G., and Osborn, J. 1994. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM J. Numer. Anal.*, 31(4):945–981. doi: 10.1137/0731051.

Brenier, Y. and Jaffré, J. 1991. Upstream differencing for multiphase flow in reservoir simulation. *SIAM J. Numer. Anal.*, 28(3):685–696. doi: 10.1137/0728036.

Chen, Z. and Hou, T. 2003. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Math. Comp.*, 72:541–576. doi: 10.1090/S0025-5718-02-01441-2.

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/>, doi: 10.2118/72469-PA.

Cusini, M., Lukyanov, A., Natvig, J., and Hajibeygi, H. 2014. A constrained pressure residual multiscale (cpr-ms) compositional solver. In *ECMOR XIV-14th European conference on the mathematics of oil recovery*, Catania, Italy. EAGE. doi: 10.3997/2214-4609.20141778.

- Efendiev, Y. and Hou, T. Y. 2009. *Multiscale Finite Element Methods*, volume 4 of *Surveys and Tutorials in the Applied Mathematical Sciences*. Springer Verlag, New York.
- Hajibeygi, H. and Jenny, P. 2009. Multiscale finite-volume method for parabolic problems arising from compressible multiphase flow in porous media. *J. Comput. Phys.*, 228(14):5129–5147. doi: 10.1016/j.jcp.2009.04.017.
- Hajibeygi, H. and Tchelepi, H. A. 2014. Compositional multiscale finite-volume formulation. *SPE J.*, 19(2):316–326. doi: 10.2118/163664-PA.
- Hou, T. and Wu, X.-H. 1997. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134:169–189. doi: 10.1006/jcph.1997.5682.
- 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. doi: 10.1016/S0021-9991(03)00075-5.
- Karypis, G. and Kumar, V. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comp.*, 20(1):359–392. doi: 10.1137/S1064827595287997.
- 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.
- Krogstad, S., Lie, K.-A., Møyner, O., Nilsen, H. M., Raynaud, X., and Skaflestad, B. 2015. Mrst-ad an open-source framework for rapid prototyping and evaluation of reservoir simulation problems. In *SPE Reservoir Simulation Symposium, 23–25 February, Houston, Texas*. doi: 10.2118/173317-MS.
- 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. In *SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February 2009*. doi: 10.2118/118993-MS.
- Krogstad, S., Lie, K.-A., and Skaflestad, B. 2012. Mixed multiscale methods for compressible flow. In *Proceedings of ECMOR XIII–13th European Conference on the Mathematics of Oil Recovery, Biarritz, France*. EAGE. doi: 10.3997/2214-4609.20143240.
- Lee, S. H., Wolfsteiner, C., and Tchelepi, H. 2008. Multiscale finite-volume formulation for multiphase flow in porous media: Black oil formulation of compressible, three phase flow with gravity. *Comput. Geosci.*, 12(3):351–366. doi: 10.1007/s10596-007-9069-3.
- Lie, K.-A., Krogstad, S., Ligaarden, I. S., Natvig, J. R., Nilsen, H., and Skaflestad, B. 2012. Open-source MATLAB implementation of consistent discretisations on complex grids. *Comput. Geosci.*, 16:297–322. doi: 10.1007/s10596-011-9244-4.
- Lipnikov, K., Moulton, J. D., and Svyatskiy, D. 2008. A multilevel multiscale mimetic ( $m^3$ ) method for two-phase flows in porous media. *J. Comput. Phys.*, 227(14):6727–6753. doi: 10.1016/j.jcp.2008.03.029.
- Lunati, I. and Jenny, P. 2006. Multiscale finite-volume method for compressible multiphase flow in porous media. *J. Comput. Phys.*, 216(2):616–636. doi: 10.1016/j.jcp.2006.01.001.
- Lunati, I. and Jenny, P. 2008. Multiscale finite-volume method for density-driven flow in porous media. *Comput. Geosci.*, 12(3):337–350. doi: 10.1007/s10596-007-9071-9.
- Lunati, I., Tyagi, M., and Lee, S. H. 2011. An iterative multiscale finite volume algorithm converging to the exact solution. *J. Comput. Phys.*, 230(5):1849–1864. doi: 10.1016/j.jcp.2010.11.036.
- Møyner, O. 2014. Construction of multiscale preconditioners on stratigraphic grids. In *ECMOR XIV – 14<sup>th</sup> European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8-11 September 2014*. EAGE. doi: 10.3997/2214-4609.20141775.
- Møyner, O. and Lie, K.-A. 2014a. The multiscale finite-volume method on stratigraphic grids. *SPE J.*, 19(5):816–831. doi: 10.2118/163649-PA.
- Møyner, O. and Lie, K.-A. 2014b. A multiscale two-point flux-approximation method. *J. Comput. Phys.*, 275:273–293. doi: 10.1016/j.jcp.2014.07.003.
- Natvig, J. R., Skaflestad, B., Bratvedt, F., Bratvedt, K., Lie, K.-A., Laptev, V., and Khataniar, S. K. 2011. Multiscale mimetic solvers for efficient streamline simulation of fractured reservoirs. *SPE J.*, 16(4):880–888. doi: 10.2018/119132-PA.
- Norne 2012. IO Center – Norne Benchmark Case. <http://www.ipt.ntnu.no/~norne>.
- Notay, Y. 2010. An aggregation-based algebraic multigrid method. *Electron. Trans. Numer. Anal.*, 37:123–140.
- Odeh, A. S. 1981. Comparison of solutions to a three-dimensional black-oil reservoir simulation problem. *J. Petrol. Techn.*, 33(1):13–25. doi: 10.2118/9723-PA.
- Pal, M., Lamine, S., Lie, K.-A., and Krogstad, S. 2013. Multiscale method for simulating two and three-phase flow in porous media. In *SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 18-20 February 2013*. SPE 163669-MS, doi: 10.2118/163669-MS.
- Sandvin, A., Keilegavlen, E., and Nordbotten, J. M. 2013. Auxiliary variables for 3d multiscale simulations in heterogeneous porous media. *J. Comput. Phys.*, 238:141–153. doi: 10.1016/j.jcp.2012.12.016.

- 
- Vanek, P., Mandel, J., and Brezina, M. 1994. Algebraic multigrid on unstructured meshes. Technical Report 34, University of Colorado at Denver, Denver, CO, USA.
- Wang, Y., Hajibeygi, H., and Tchelepi, H. A. 2014. Algebraic multiscale solver for flow in heterogeneous porous media. *J. Comput. Phys.*, 259:284–303. doi: 10.1016/j.jcp.2013.11.024.
- Zhou, H. and Tchelepi, H. A. 2008. Operator-based multiscale method for compressible flow. *SPE J.*, 13(2):267–273. doi: 10.2118/106254-PA.
- Zhou, H. and Tchelepi, H. A. 2012. Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models. *SPE J.*, 17(2):523–539. doi: 10.2118/141473-PA.