5

Applications of multiscale finite element methods

5.1 Introduction

In this chapter, we present some applications of MsFEM to fluid flows in heterogeneous porous media. We discuss multiscale methods for transport equations and their coupling to flow equations which are solved using MsFEMs. The proposed multiscale techniques for the transport equation share some similarities with nonlinear multiscale methods introduced in Chapter 3. Because of sharp interfaces, special treatment is needed near the interface. Furthermore, due to the hyperbolic nature of the transport equation, some type of limited global information is needed for constructing multiscale basis functions. These issues are discussed in Section 5.2.

In Section 5.3, we discuss the applications of MsFEMs to flows in unsaturated porous media described by Richards' equations [236]. Multiscale methods developed in Chapter 3 are applied to solve Richards' equation in heterogeneous porous formations on the coarse grid. In Section 5.4, we extend MsFEMs to solving the fluid-structure problem on the coarse grid where as a result of fluid flow in the pore region, the porous medium deforms substantially.

Applications of MsFEMs to reservoir modeling are presented using both the mixed MsFEM and MsFV in Sections 5.5 and 5.6. In these sections, more complicated porous medium equations involving compressibility, gravity, and three phases in heterogeneous reservoirs are considered. The authors address the challenging issues that arise in petroleum applications and describe the efficient use of MsFEMs in these problems.

The porous medium properties are typically described using geostatistical techniques because of uncertainties associated with prescribing permeability values to different locations. The numerical simulation of fluid flows in stochastic porous media is prohibitively expensive because the computation of each realization is CPU-demanding. In this chapter, we also consider approaches for constructing multiscale basis functions for the whole ensemble. Furthermore, the applications of MsFEMs to uncertainty quantification in inverse problems

Y. Efendiev, T.Y. Hou, *Multiscale Finite Element Methods: Theory and Applications,*
Surveys and Tutorials in the Applied Mathematical Sciences 4,
DOI 10.1007/978-0-387-09496-0_5, © Springer Science+Business Media LLC 2009

consisting of permeability sampling are presented. The objective here is to use MsFEMs to speedup the computations aimed at quantifying uncertainties in inverse problems.

5.2 Multiscale methods for transport equation

5.2.1 Governing equations

A prototypical example for problems studied is two-phase immiscible flow and transport in heterogeneous media. We presented the governing equations in Section 2.10 neglecting the effects of gravity, compressibility, capillary pressure and dispersion on the fine scale. We recall that the system of equations consists of the pressure equation

$$\mathrm{div}(\lambda(S)k(x)\nabla p) = q_t, \tag{5.1}$$

where $\lambda(S)$ is the total mobility and $q_t = q_o + q_w$ is the total volumetric source term. The saturation equation has the form

$$\phi\frac{\partial S}{\partial t} + \mathrm{div}(vf(S)) = -q_w, \tag{5.2}$$

where $f(S)$ is the fractional flow of water (f is also denoted by f_w often to distinguish between oil and water fractional flows), and ϕ is porosity. The total velocity v is given by

$$v = -\lambda(S)k\nabla p. \tag{5.3}$$

In the presence of capillary effects, an additional diffusion term is present in (5.2). The above system of equations can be extended to describe the flow and transport of three-phase flow and transport (see Sections 5.5 and 5.6). In Sections 5.5 and 5.6, the applications of MsFEMs to three-phase compressible flow and transport are described.

In this section, we focus on developing multiscale methods for the transport equation described by (5.2).

5.2.2 Adaptive multiscale algorithm for transport equation

In this section, we present an adaptive multiscale method for solving the transport equation following [5]. The main idea of this approach is to construct multiscale basis functions similar to the construction in nonlinear MsFEMs presented in Chapter 3. Because the solution of the transport equation has sharp interfaces, a separate treatment is needed for these interfaces.

The adaptive multiscale method that we propose here consists of two parts. An adaptive criterion determines if a block is in a transient flow region. Here, by transient region, we refer to those regions with sharp saturation fronts. In these regions we use local fine-grid computations to advance the saturation solution to the next time-step. In regions with slow transients, we use a multiscale coarse-grid solver to advance the saturation solution to the next time-step. Then, instead of doing a fine-grid calculation, we map the coarse-grid solution onto a fine-grid solution using special interpolation operators.

Before we give an outline of the algorithm, we need to introduce some additional notation. First, denote the coarse grid by $\mathcal{T} = \{K_i\}$ and an underlying fine grid by $\mathcal{K} = \{\tau_i\}$. The grids used here need not coincide with the coarse and fine grids for multiscale methods used for the pressure equation and can be unstructured. In this particular application, we use a mixed MsFEM.

We introduce now the upstream fractional flow function for $\gamma_{ij} = \partial K_i \bigcap \partial K_j$:

$$V_{ij}(S) = f(S_i)\max\{v_{ij}, 0\} + f(S_i)\min\{v_{ij}, 0\}, \tag{5.4}$$

where v_{ij} is the Darcy flux across γ_{ij} that we get from the mixed MsFEM solution. Next, let $\bar{S}_i^n$ be the coarse-grid saturation in K_i at time t_n, and denote by $\mathcal{T}_{\mathrm{tr}}^n$ the family of grid blocks that are identified to be in a transient flow region at time t_n. One can use various criteria based on coarse-scale saturation values or their gradients to identify transient regions. In this section, the following criteria are used to identify transient flow regions:

$$K_i \in \mathcal{T}_{\mathrm{tr}}^n \ \text{ if } \ \max\{|\bar{S}_i^n - \bar{S}_j^n| : |\partial K_i \cap \partial K_j| > 0\} \geq \alpha_i. \tag{5.5}$$

For each $K_i \in \mathcal{T}_{\mathrm{tr}}^n$, we define

$$K_i^E = K_i \cup \{\tau \in \mathcal{K} : |\partial\tau \cap \partial K_i| > 0\}.$$

Hence, K_i^E consists of grid cells that are either contained in K_i, or that share a common interface with a cell in K_i. Finally, we introduce a family of operators $\{I_K : K \in \mathcal{T}\}$ that map coarse-grid saturations onto fine-grid saturation fields inside the respective blocks. The adaptive multiscale method is now outlined in Algorithm 5.2.1.

Next, we briefly describe the algorithm. In this algorithm, first, the fine-grid saturations in the transient flow regions are updated. This update involves solving the local transport equation on the fine grid in the transient region. Coarse-grid saturations in nontransient regions are updated using (5.7). The equation (5.7) is obtained by averaging the transport equation over the coarse-grid block K and describes the update for the coarse-scale saturation field. Once the coarse-scale saturation field is updated, it is mapped onto the fine grid with the coarse-to-fine grid interpolation operators. This step is similar to nonlinear MsFEMs as described in Section 3.1. In particular, the basis functions are computed for different levels of average saturation within the coarse grid block and, then, interpolated. In the algorithm, implicit time integration methods are used. There are no constraints on the time-steps $\triangle t$, but they should be chosen small enough to avoid an excessive numerical diffusion.

Algorithm 5.2.1 Adaptive multiscale algorithm for modeling flow in porous media

```
For each $K \in \mathcal{T}_{\rm tr}^n$, do
  – For $\tau_i \subset K^E$, compute

      $$S_i^{n+1/2} = S_i^n + \frac{\triangle t}{\int_{\tau_i} \phi \, dx} \left[ \int_{\tau_i} -q_w(S^{n+1/2}) dx - \sum_{j \neq i} V_{ij}^* \right], \qquad (5.6)$$

    where $V_{ij}^* = \begin{cases} V_{ij}(S^n) & \text{if } \gamma_{ij} \subset \partial K^E \text{ and } v_{ij} < 0. \\ V_{ij}(S^{n+1/2}) & \text{otherwise.} \end{cases}$
  – Set $S^{n+1}|_K = S^{n+1/2}|_K$.

For each $K \notin \mathcal{T}_{\rm tr}^n$, do
  – Set $S^{n+1}|_K = S^n|_K$.
  – While $\sum_j \triangle_j t \leq \triangle t$, compute

      $$\bar{S}_K^{n+1} = \bar{S}_K^n + \frac{\triangle_j t}{\int_K \phi \, dx} \left[ \int_K -q_w(S^{n+1}) \, dx - \sum_{\gamma_{ij} \subset \partial K} V_{ij}(S^{n+1}) \right], \qquad (5.7)$$

    and set $S^{n+1}|_K = I_K(\bar{S}_K^{n+1})$.
```

The fractional function f is in general a nonlinear function of saturation. We therefore solve the fine-grid equations (5.6) using a Newton–Raphson method. Here saturation from the previous time-step is used to determine boundary conditions along the inflow boundary on ∂K^E. This gives rise to a mass–balance error because the inflow on grid block boundaries corresponding to the saturation from the previous time-step will not match exactly the inflow on grid block boundaries corresponding to the saturation at the current time-step. In our numerical simulations, we observed that this mass–balance error is usually very small, and generally insignificant. Note also that if $\mathcal{T}_{\rm tr} = \emptyset$, and the coarse-to-fine grid interpolation conserves mass locally, then (5.7) ensures that mass is conserved, also globally. Thus, under the assumption that the coarse-to-fine grid interpolation conserves mass locally, the latter part of the adaptive multiscale algorithm is mass conservative on both coarse and fine grids.

Next, observe that fluxes across coarse-grid interfaces in (5.7) are evaluated on fine-grid interfaces $\gamma_{ij} \subset \partial K$. Thus, rather than using a flux function that models the total flux across coarse-grid interfaces as a function of the net saturation in the upstream block, we evaluate the term fv in (5.2) on the scale of the fine grid. This requires that we have fine-grid saturation values in all cells adjacent to grid block boundaries. The coarse-to-fine grid interpolation operators $\{I_K\}$ are therefore not just tools to get better resolution. In addition to improving the global accuracy of Algorithm 5.2.1 by providing a better approximation to flow across coarse-grid interfaces, they provide initial fine-grid saturation values for (5.6) in the transition when a block is identified as being part of a transient flow region. Without the interpolation, the initial saturation field for (5.6) would be constant in K, and the fractional flow across the coarse-grid interfaces would have to be based on the net grid block saturations only, as pseudo-functions generally do [171].

We remark that the proposed adaptive multiscale method has some similarities to the multiscale framework developed for nonlinear equations in which multiscale basis functions are constructed by mapping the coarse dimensional space defined over the entire region. Furthermore, this map is used in the global coarse-grid formulation of the fine-scale problem to compute the coarse-scale solution. In our multiscale approach, the basis functions are constructed as a function of average saturation in each coarse block, and then used in the global formulation of the problem. In both approaches, the main task is to determine an accurate and efficient multiscale map that improves the global coarse-grid formulation of the problem.

5.2.3 The coarse-to-fine grid interpolation operator

In the following we attempt to construct operators that map each coarse-grid saturation field onto a fine-scale saturation profile that is close to the corresponding profile that one would get by solving the saturation equation on the global fine grid. The basic idea is to approximate the fine-scale saturation in K_i as a linear combination of two basis functions Φ_i^k and Φ_i^{k+1} with $\int_{K_i} \Phi_i^k \phi \, dx \leq \bar{S}_i^n \int_{K_i} \phi \, dx < \int_{K_i} \Phi_i^{k+1} \phi \, dx$:

$$I_{K_i}(\bar{S}_i^n) = \eta \Phi_i^k + (1-\eta)\Phi_i^{k+1}. \tag{5.8}$$

Here $\eta \in [0,1]$ is chosen such that the interpolation preserves mass, that is such that

$$\int_{K_i} I_{K_i}(\bar{S}_i^n)\phi \, dx = \bar{S}_i^n \int_{K_i} \phi \, dx. \tag{5.9}$$

This condition states that the fluid contained in K_i is distributed inside K_i in such a way that the total fluid volume in K_i is conserved. The basis functions $\Phi_i^k = s_i(x, \tau_k)$ represent snapshots of the solution of the following equation:

$$\phi \frac{\partial s_i}{\partial t} + \operatorname{div}(f(s_i)v) = -q_w \ \text{ in } K_i. \tag{5.10}$$

For the local problem (5.10) to be well defined, we need to specify initial conditions and boundary conditions, and provide a possibly time-varying velocity field in K_i. Unfortunately, we do not know a priori what the velocity will be during the simulation, nor what boundary conditions to impose. Assumptions must therefore be made as to how the velocity and saturation approximately evolve. We describe below an approach that is local in terms of boundary and initial conditions, however, one can naturally incorporate global information into this approach. The proposed approach assumes that global boundary conditions for the pressure equation (5.1) are not changed, and that the source terms are fixed. We assume also that an upstream method is used to solve the local equations (5.10). Thus, we need only specify boundary conditions on the inflow boundaries $\Gamma_K^{\text{in}} = \{\gamma_{jl} \subset \partial K : \tau_l \subset K,\ v_{jl} < 0\}$.

For fixed flow conditions, the fine-scale velocity features will generally not change significantly during a flow simulation. This is discussed in [5]. One option is therefore to solve the pressure equation (5.1) at the initial time with the mixed MsFEM, use $v = v(x, t_0)|_K$ in (5.10), and the same initial data as for the global problem (5.2). A local way of generating saturation basis functions based on this approach requires that sensible boundary conditions for (5.10) can be imposed for each block independently. In our numerical simulations, we impose $s_i = 1$ on the inflow boundary Γ_T^{in}, although other boundary conditions can be imposed (see discussions in Section 5.2.7).

An approach that is often used in practice for upscaling the saturation equation entails the use of so-called pseudo-relative permeabilities $(k_{rj}^*)_i = (k_{rj}^*)_{K_i}$ in place of the fine-scale k_{rj}. Because the fine-scale k_{rj} are typically functions only of saturation S, pseudo-relative permeabilities, or pseudo-functions for brevity, are commonly assumed to depend only on the coarse-grid saturation $\overline{S}$, though the curves can vary between coarse grid blocks. The proposed technique shares some similarities with pseudo-function approaches although there are some important differences. The proposed approach allows recovering fine-scale features of the saturation field and can be used for accurate upscaling. The relation between proposed methods and pseudo-function approaches is discussed in [5].

5.2.4 Numerical results

We now use the proposed methodology to model incompressible and immiscible two-phase flow on test cases with permeability and porosity from SPE 10 [78]. This model was discussed before and consists of a Tarbert formation on top of a fluvial upper Ness formation. Although both formations are very heterogeneous, the upper Ness formation gives rise to more complex flow. We employ here mostly data modeling parts of the fluvial upper Ness formation. Because fluvial formations are particularly hard to upscale, the upper Ness formation should serve as an appropriate model for testing and validation of the proposed multiscale method. The upper Ness model is Cartesian and consists of $60 \times 220 \times 50 = 6.6 \cdot 10^5$ grid cells.

We assume that the reservoir is initially fully oil-saturated, and inject water at a constant rate in grid cells penetrated by a vertical well at the center of the domain. We then produce at the producers which are vertical wells located at each of the four corners. The water and oil mobilities are defined by

$$\lambda_w(S) = \frac{S^2}{\mu_w} \text{ and } \lambda_o(S) = \frac{(1-S)^2}{\mu_o}, \tag{5.11}$$

where the water and oil viscosities are assumed to be equal: $\mu_w = \mu_o = 0.003$ cp.

To measure the overall accuracy of a saturation solution we compute the error in the fine- and coarse-grid saturation profiles relative to a reference solution,

$$e(S, S_{\text{ref}}, t) = \frac{\|\phi S_{\text{ref}}(\cdot, t) - \phi S(\cdot, t)\|_{L^2}}{\|\phi S_{\text{ref}}(\cdot, t) - \phi S_{\text{ref}}(\cdot, 0)\|_{L^2}}.$$

Here time is measured in dimensionless time PVI, that is time measures the fraction of the total accessible pore volume in Ω that has been injected into Ω.

For all test cases, we use Cartesian coarse grids, and assume that the fine-grid cells coincide with grid cells in the original Cartesian grid. The reference solution S_{ref} is computed using an implicit upstream method on the fine grid, and a corresponding coarse-grid solution is computed using the same method on a coarse grid. Moreover, note that although we use a fixed set of basis functions for the mixed MsFEM, we solve the pressure equation repeatedly to account for mobility variations. Thus, the velocity fields in the simulations will differ from the velocity field used to generate the saturation basis functions. However, to assess the accuracy of solutions obtained using the adaptive multiscale algorithm (AMsA), we compute, at each pressure time-step, the velocity field corresponding to the reference solution for saturation, and use this velocity field in AMsA, and to compute the coarse-grid solution. This allows us to monitor the error that stems from AMsA only.

5.2.5 Results for a two-dimensional test case

We consider first a test case representing the bottom layer of the SPE model. The coarse grid is defined so that each grid block contains 10 × 10 grid cells. The saturation plots in Figure 5.1 show that the solutions obtained using AMsA with $\alpha = 0$, $\alpha = 0.1$, and $\alpha = 0.2$ (the same threshold is used in all grid blocks, see (5.5)) are very similar to the reference solution. We recall that $\alpha = 0$ corresponds to the case when the saturation update is performed in all coarse blocks and $\alpha = 1$ corresponds to the case when no saturation update is performed. The solution obtained using $\alpha = 1$ looks quite different compared to the cases with other values of α. The sharp edges that we see in this plot are due to the fact that the boundary conditions used to generate the saturation basis functions overestimate the inflow. We therefore get too much saturation along the inflow part (with respect to the initial velocity field) of each grid block boundary. This indicates that without the adaptive component, AMsA is not able to provide plausible fine-grid saturation profiles. To achieve this, one has to build more information about the global flow problem into the saturation basis functions by specifying appropriate coarse grid blocks using global information or appropriate dynamic boundary conditions for (5.10).

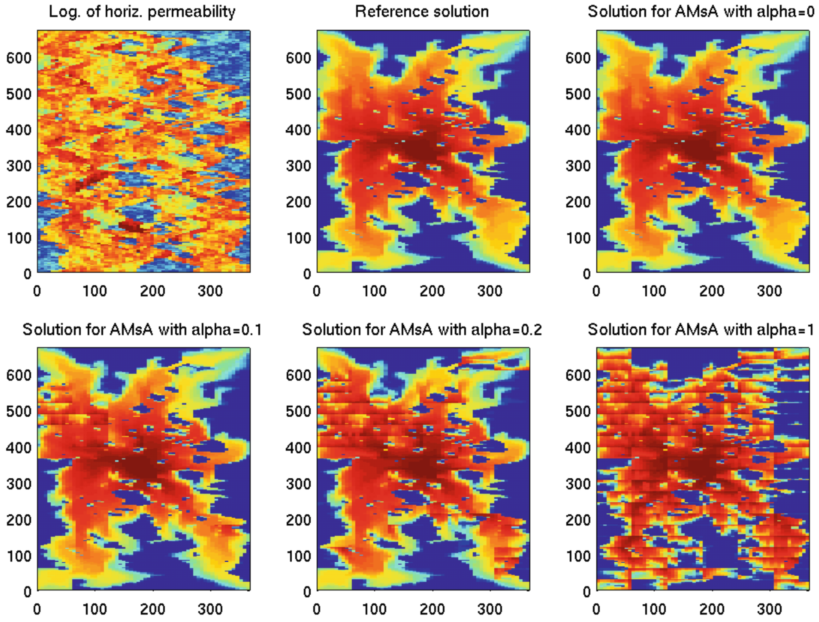


Fig. 5.1. Saturation profiles at ~ 0.7 PVI for simulations on the bottom layer.

Figure 5.2 shows that the accuracy of AMsA decays with increasing α. However, for all α, AMsA gives a significantly more accurate solution on the coarse grid than the standard upstream method on the coarse grid gives.

Computational efficiency

Except for $\alpha = 1$, for which local problems are not solved during the course of a flow simulation, the computational cost of AMsA is dominated by the cost of solving the local equations (5.6). In particular, for small α the computational cost $C(\alpha)$ of solving (5.2) using AMsA scales roughly as

$$C(\alpha) \sim F_u(\alpha) N_t C(0),$$

where N_t is the total number of time-steps and $F_u(\alpha)$ is the average fraction of blocks that belong to a transient flow region. Note that $C(0)$ is the cost when the transient region is the entire domain.

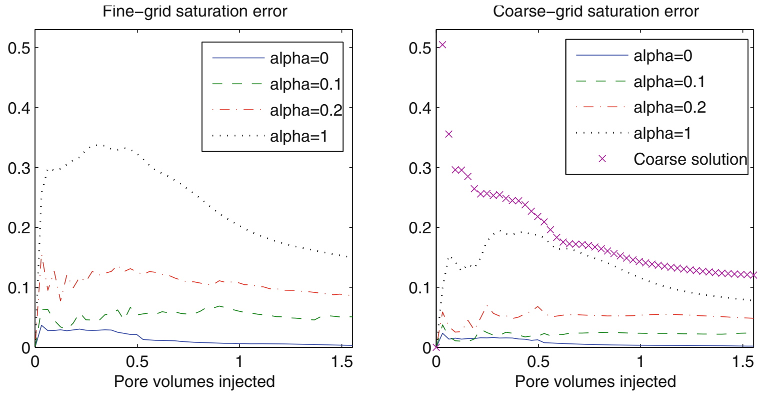


Fig. 5.2. Saturation errors for saturation solutions obtained from simulations on the bottom layer of the upper Ness formation. The fine-grid curves measure the error with $e(S, S_{\text{ref}}, t)$ on the fine grid relative to the reference solution, and the coarse-grid curves measure the error on a coarse grid with $e(\bar{S}, \bar{S}_{\text{ref}}, t)$ relative to the projection of the reference solution onto the coarse grid.

Clearly, F_u is a decreasing function of α. Hence, there is a trade-off between high accuracy and low computational cost. Note also that, in addition to α, F_u depends implicitly on various factors (e.g., the coarse grid, the criteria used to identify transient flow regions, the fluid parameters, the heterogeneous structures, etc.). In particular, AMsA is in general more efficient (and accurate) for spatially correlated variogram-based permeability models than for models with fluvial heterogeneity, as is illustrated in Figure 5.3. Whereas, on average, 73% and 55% of the blocks in the upper Ness model are identified as belonging to transient flow regions for $\alpha = 0.1$ and $\alpha = 0.2$ respectively, the corresponding numbers for the Tarbert model are 46 and 27. The potential efficiency of AMsA is therefore highly dependent on the type of model to which it is applied. Relative to AMsA with $\alpha = 0$, we may expect good accuracy on both coarse and fine grids, with a speed-up factor about two for models with fluvial heterogeneity, and a speed-up factor three or four for models with smoother heterogeneity. The speed-up strongly depends on the adaptivity criteria which can be adjusted for a particular problem. In our simulations, the criteria based on gradients of the coarse-scale saturation are used. We have observed an increase in speed-up when the criteria based on saturation values are used. Without the adaptive component, the computational complexity of AMsA is comparable to the complexity of coarse-grid simulations using pseudo-functions. As we mentioned earlier, the accuracy of AMsA can be improved by choosing adaptive coarse gridding. This procedure will also enhance the efficiency of AMsA, because it localizes sharp fronts. Finally, we note that the purpose of the interpolator is not primarily to get the fine-scale details correct, but rather to introduce a flexible mechanism that allows us to capture the subgrid transport effects on a coarse scale.

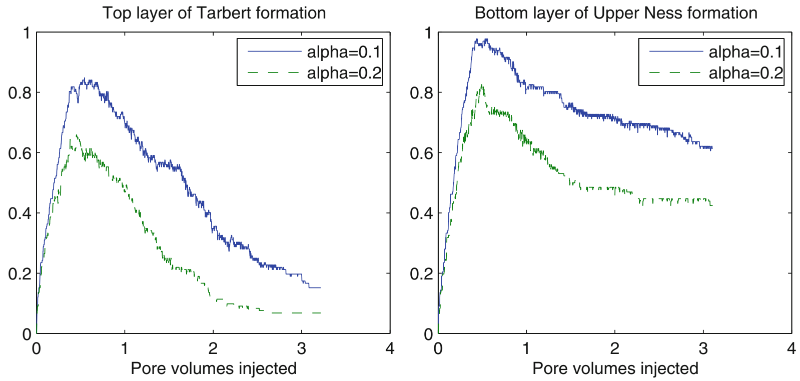


Fig. 5.3. Fraction of blocks that are identified to belong to transient flow regions during the course of two-phase flow simulations on the top layer of the Tarbert formation (left) and the bottom layer of the upper Ness formation (right).

5.2.6 Three-dimensional test cases

In this section we want to examine the accuracy of AMsA when applied to two-phase flow simulations on three-dimensional models from the upper Ness formation. Here we consider only AMsA using $\alpha = 0$, $\alpha = 0.1$, and $\alpha = 1$ in all blocks. The case $\alpha = 0$ is referred to as the domain decomposition (DD) algorithm, the case $\alpha = 0.1$ is referred to as the adaptive algorithm, and the case $\alpha = 1$ is called the multiscale algorithm.

In order for AMsA to provide a valuable tool in reservoir simulation, it should, in addition to being significantly more accurate than the coarse-grid solution, capture fine-scale characteristics of the reference solution at well locations. This is demonstrated by comparing water-cut curves (fraction of water in the produced fluid) for AMsA with water-cut curves for the reference solution. To get accurate production characteristics, it is essential that high-flow channels are resolved adequately because high-flow channels often carry the majority of the flow that reaches the producers. Thus, if AMsA can be used to model these regions properly, then they should provide a more robust alternative to reservoir simulation on upscaled models.

Consider first the ten bottom layers of the upper Ness formation, and define the coarse grid so that each grid block in the coarse grid consists of $10 \times 10 \times 5$ grid cells. Figures 5.4 and 5.5 demonstrate that all AMsAs give significantly more accurate results than the solution obtained by solving the saturation equation on the coarse grid with the implicit upstream method. We notice, in particular, that the water-cut curves for the multiscale algorithm are much more accurate than the corresponding water-cut curves for the coarse-grid solution. This indicates that AMsA is more capable of resolving high-flow regions adequately, also without the local fine-grid computations.

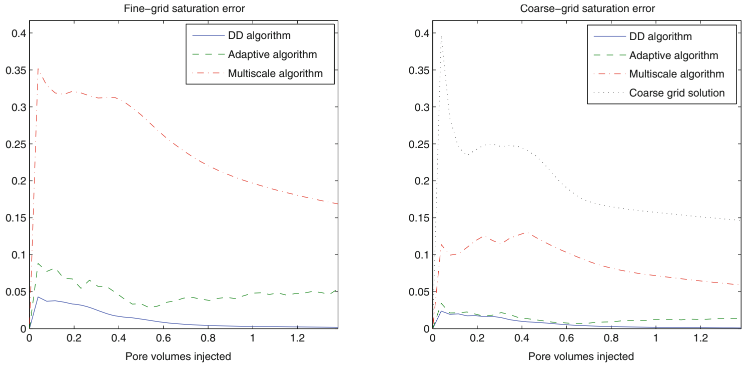


Fig. 5.4. Saturation errors for simulations on the bottom ten layers of the upper Ness formation.

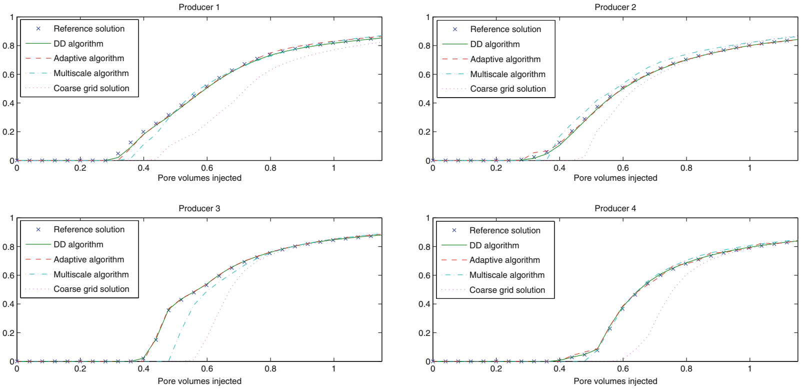


Fig. 5.5. Water-cut curves for simulations on the bottom ten layers of the upper Ness formation.

We turn now to the full three-dimensional model of the upper Ness formation. The previous examples showed that the DD algorithm seems to produce solutions that very closely match the reference solution, and it is computationally very expensive to compute a solution on the full upper Ness model using the implicit upstream method on the fine grid, therefore we use here the solution obtained using the DD algorithm as the reference solution. Again we let the coarse grid be defined so that each grid block in the grid consists of $10 \times 10 \times 5$ grid cells.

Figure 5.6 demonstrates that the errors are approximately the same as in the previous example. We observe also that the saturation error on the coarse grid for the multiscale algorithm is less than half of the corresponding error for the coarse-grid solution. Furthermore, the water-cut curves for the multiscale algorithm depicted in Figure 5.7 closely match the water-cut curves for the adaptive algorithm and the DD algorithm, except possibly for producer 4 where we observe a mismatch. In contrast, the coarse-grid solution continues to overestimate the breakthrough times, and thus overpredicts the oil production. This shows that the multiscale method may be used as an alternative to pseudo-functions for enhancing the accuracy of coarse-grid simulations.

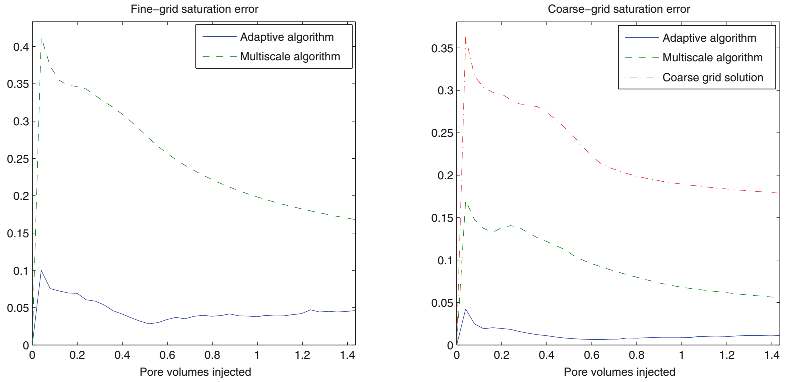


Fig. 5.6. Saturation errors for simulations on the full upper Ness formation.

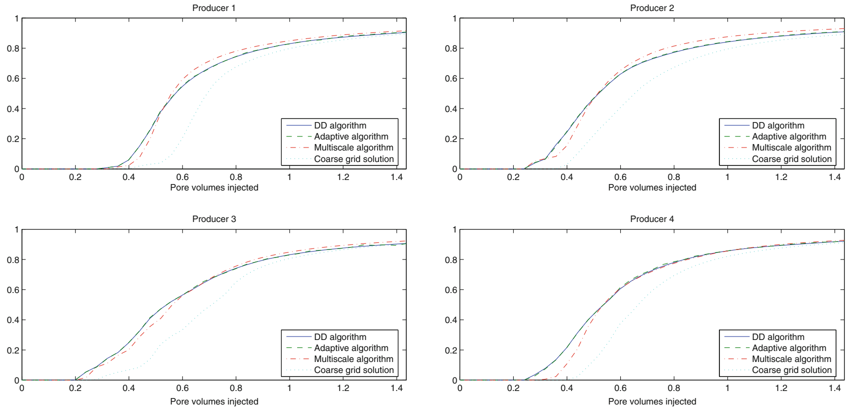


Fig. 5.7. Water-cut curves for simulations on the full model of the upper Ness formation.

5.2.7 Discussion on local boundary conditions

In our numerical simulations, we imposed $s_i = 1$ on the inflow boundary Γ_T^{in} (see (5.10)). For general coarse grids, these boundary conditions may seem a bit crude. Indeed, these boundary conditions are exact only if there is a sharp front in the global solution that, for each block, hits the whole inflow boundary at approximately the same instant. It should be emphasized that the purpose of the interpolator is not primarily to get the fine-scale details correct, but rather to introduce a flexible mechanism that allows us to model the flow on a coarse scale more correctly.

To get accurate solutions, also on fine grids, one must either use an adaptive component to improve the solution in transient flow regions, or build more information into the interpolator. For instance, note that the inherent flexibility with respect to coarse grids allows us to reduce the error associated with this type of boundary condition by using flow-based, non-Cartesian grids. In particular, by using coarse blocks with boundaries aligned with level sets of time-of-flight function, one can achieve higher accuracy compared to the approaches where Cartesian coarse blocks (or coarse blocks selected independent of global flow features) are used. This option is discussed in [5]. We note that our numerical results show that the multiscale approach using Cartesian coarse blocks still provides a good overall accuracy. One can also use limited global information, such as the time-of-flight function, in constructing coarse blocks.

5.2.8 Other approaches for coarsening the transport equation

There are a number of other techniques for coarsening the saturation equation that can be coupled to the pressure equation. Next, we describe a few of these approaches very briefly without detailed numerical studies which can be found in the literature.

A macrodispersion model for transport equation

The approach entails using a macrodispersion formulation for the coarse-scale saturation equation. We consider the upscaling of the saturation equation using perturbation techniques following, for example, [102, 101]. We omit the details of the derivation of the upscaled model. We first consider the case $\lambda(S) = 1$ and $f(S) = S$ in (5.1) and (5.2) (with $q_w = 0$ in (2.41)). The upscaled model was derived using perturbation arguments for (5.2), in which the saturation S and the velocity v on the fine scale are assumed to be the sum of their volume-averaged and fluctuating components,

$$v = \overline{v} + v', \quad S = \overline{S} + S'. \tag{5.12}$$

Here the overbar quantities designate the volume average of fine-scale quantities over coarse blocks. For simplicity, one can assume that the coarse blocks are rectangular which allows stating $\overline{\nabla F} = \nabla \overline{F}$, if averages are taken over dual volume. In general, one can also perform the perturbation technique directly on the target coarse block as done in [102]. In this case, the averages of divergences can be written over the boundaries of the coarse blocks. Substituting (5.12) into the saturation equation for single-phase and averaging over coarse blocks we obtain

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla \overline{S} + \overline{v' \cdot \nabla S'} = 0. \tag{5.13}$$

The term $\overline{v' \cdot \nabla S'}$ represents subgrid effects due to the heterogeneities of convection. This term can be modeled using the equation for $S^{'}$ that is derived by subtracting (5.13) from the fine-scale equation (5.2),

$$\frac{\partial S^{'}}{\partial t} + \overline{v} \cdot \nabla S^{'} + v^{'} \cdot \nabla \overline{S} + v^{'} \cdot \nabla S^{'} = \overline{v' \cdot \nabla S'}.$$

This equation can be solved along the characteristics $dx/dt = \overline{v}$ by neglecting higher-order terms. Carrying out the calculations in an analogous manner to the ones performed in [102] we can easily obtain the following coarse-scale saturation equation

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla \overline{S} = \mathrm{div}(D(x,t) \nabla \overline{S}(x,t)), \tag{5.14}$$

where $D(x,t)$ is the dispersive matrix coefficient, whose entries are written as

$$D_{ij}(x,t) = \left[\int_0^t \overline{v_i'(x) v_j'(x(\tau))} d\tau \right]. \tag{5.15}$$

Next it can be easily shown that the diffusion coefficient can be approximated up to the first order by

$$D_{ij}(x,t) = \overline{v_i'(x) L_j^D}$$

where L_j^D is the displacement of the particle in the j direction that starts at the point x and travels with velocity $-v$. The diffusion term in the coarse model for the saturation field (5.14) represents the effects of the small scales on the large ones. Note that the diffusion coefficient is a correlation between the velocity perturbation and the displacement. This is different from [102] where the diffusion is taken to be proportional to the length of the coarse-scale trajectory. Using MsFEMs for the pressure equation we can recover the small-scale features of the velocity field that allow us to compute the fine-scale displacement.

For the nonlinear flux $f(S)$, we can use a similar argument by expanding $f(S) = f(\overline{S}) + f_S(\overline{S}) S^{'} + \cdots$. In this expansion we take into account only linear terms and assume that the flux is nearly linear. This case is similar to the linear case and the analysis can be carried out in an analogous manner. The resulting coarse-scale equation has the form

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla \overline{S} = \mathrm{div}(f_S(\overline{S})^2 D(x,t) \nabla \overline{S}(x,t)), \tag{5.16}$$

where $D(x,t)$ is the macrodiffusion corresponding to the linear flow. This formulation has been derived within the stochastic framework in [173]. We note that the higher-order terms in the expansion of $f(S)$ may result in other effects that have not been studied extensively to our best knowledge. In [101] the authors use a similar formulation although their implementation is different from ours. Numerical results can be found in [102, 101].

Coarsening in a flow-based coordinate system

In [106, 247], a flow-based coordinate system is used to coarsen the saturation equation. A flow-based coordinate system consists of single-phase pressure and the corresponding streamfunction fields. The use of global information can improve the multiscale finite element method. In particular, the solution of the pressure equation at the initial time is used to construct the boundary conditions for the basis functions. It is interesting to note that the multiscale finite element methods that employ limited global information reduce to the standard multiscale finite element method in a flow-based coordinate system. This can be verified directly and the reason behind it is that we have already employed limited global information in a flow-based coordinate system.

To achieve a high degree of speedup in two-phase flow computations, we consider the upscaling of the transport equation in a flow-based coordinate system. Flow-based coordinate systems simplify the scale interaction and allow us to perform upscaling of the transport equation. In particular, in a flow-based coordinate system, the saturation equation becomes one-dimensional with a varying velocity field along the streamlines. This allows us to use the perturbation approach and perform upscaling using macrodispersion models.

Extensive numerical studies are presented in [247, 106]. These numerical tests use the MsFVEM for two-phase flow. Note that global information is already incorporated into the multiscale basis functions and the standard MsFVEM is equivalent to the MsFVEM using limited global information introduced earlier. In our simulations, a moving mesh is used to concentrate the points of computation near the sharp front. Because the saturation equation is one-dimensional in the pressure–streamline coordinates, the implementation of the moving mesh is straightforward and efficient. We have presented the numerical results for different types of heterogeneities. All numerical results show that one can achieve accurate results with low computational cost.

Multiscale analysis for convection dominated equations

In this section, we consider a systematic upscaling framework for the transport equation based on multiscale homogenization. In [144], Hou, Westhead, and Yang introduced a novel multiscale analysis for the two-phase immiscible flows in heterogeneous porous media. In particular they derived the homogenized equations by projecting the fluctuation of saturation onto a suitable subspace. Furthermore, they demonstrated by extensive numerical experiments that the upscaling method can accurately capture the multiscale solution of the two-phase flow. Very recently, Hou and Liang [142] further improved the multiscale analysis of Hou et al. [144] and developed a systematic multiscale analysis to upscale convection-dominated transport equations.

To demonstrate the main idea, we consider the following transport equation which contains a strong convection term and a weak diffusion term

$$\frac{\partial S_\epsilon}{\partial t} + v(x, \frac{x}{\epsilon}, t) \cdot \nabla S_\epsilon = \epsilon^m \mathrm{div}(D(x, \frac{x}{\epsilon}, t)\nabla S_\epsilon),$$

where $S_\epsilon|_{t=0} = S_I(x)$, $m \in [2, \infty]$ is an integer, $v(x, y, t)$ and $D(x, y, t)$ are assumed to be periodic in $y = x/\epsilon$, and ϵ characterizes the small scale in the media, Moreover, we assume that v is oscillatory divergence-free with respect to the fast variable y; that is $div_y(v) = 0$. The local Peclet number is of order $O(\epsilon^{-m+1})$.

Next, we define a null space $\mathcal{N}$, $\mathcal{N} = \{f \in H^1_Y, v \cdot \nabla_y f = 0, \forall y \in Y\} \subset L^2_Y$, where L^2_Y is the L^2 space of periodic functions. This functional space plays an important role in our multiscale analysis. We also introduce a range space $\mathcal{W}$, $\mathcal{W} = \{v \cdot \nabla_y \theta : \theta \in H^1_Y\}$. In [144], the authors have shown that $\mathcal{N}$ and $\mathcal{W}$ form an orthogonal decomposition of L^2_Y; that is

$$L^2_Y = \mathcal{N} \oplus \overline{\mathcal{W}}.$$

Let $\mathcal{P}$ be the projection $H^1_Y \to \mathcal{N}$. Define the projection $\mathcal{Q}$: $L^2_Y \to \mathcal{W}$ as

$$\|g - \mathcal{Q}(g)\| = \min_{\theta \in H^1_Y} \|g - v \cdot \nabla_y \theta\|.$$

As pointed out in [144], $\mathcal{P}$ is related to $\mathcal{Q}$ via $\mathcal{P}(g) = g - \mathcal{Q}(g)$, and $\mathcal{Q}$ can be computed by $\mathcal{Q}(g) = v \cdot \nabla_y \theta$, where θ is the solution of

$$\mathrm{div}_y(E\nabla_y \theta) = v \cdot \nabla_y g, \quad y \in Y, \tag{5.17}$$

with periodic boundary condition and the matrix is defined by $E = v^T v$ whose (i, j) entry is given by $v_i v_j$, where $v = (v_1, v_2, v_3)$. Moreover, the projection operator $\mathcal{P}$ is equivalent to the streamline averaging projection operator [144, 142].

Guided by our multiscale analysis, we look for a multiscale expansion of the concentration in the form

$$S_\epsilon(x, t) = S_0(x, x/\epsilon, t) + \epsilon S_1(x, x/\epsilon, t) + O(\epsilon^2),$$

where S_j $(j = 0, 1)$ are periodic functions of y.

In [142], we showed that the leading-order approximation S_0 satisfies the following homogenized equations

$$v \cdot \nabla_y S_0 = 0, \tag{5.18}$$

$$\frac{\partial S_0}{\partial t} + v \cdot \nabla_x S_0 + v \cdot \nabla_y w = 0, \tag{5.19}$$

where $w \in \mathcal{W}$, and the initial condition is given by $S_0|_{t=0} = S_I(x)$.

Note that there are two equations for S_0 given by (5.18) and (5.19), but there is no evolution equation for w. The equation for w can be derived by imposing the algebraic constraint (5.18) for S_0. The role of w is to enforce $v \cdot \nabla_y S_0 = 0$, which is similar to the role that the pressure plays in the incompressible Navier–Stokes equations. The solution w can be obtained by solving (5.17). In [144], an effective iterative method was introduced to solve the degenerate elliptic equation (5.17).

One of the main contributions of [142] is to show that the homogenized equations (5.18) and (5.19) are well-posed and obtain an optimal error estimate

$$\|S_\epsilon(x,t) - S_0(x, \frac{x}{\epsilon}, t)\|_{L^2} \leq C\epsilon.$$

We now decompose c_0 and v into the sum of their average and fluctuation, $S_0(x,y,t) = \overline{S_0}(x,t) + S_0'(x,y,t)$, $v(x,y,t) = \overline{v}(x,t) + v'(x,y,t)$, where $\overline{f}(x,t) = \int_Y f(x,y,t)dy$. It is easy to show that $\overline{S_0}$ and S_0' satisfy the following equations

$$\frac{\partial \overline{S_0}}{\partial t} + \overline{v} \cdot \nabla_x \overline{S_0} + \overline{v' \cdot \nabla_x S_0'} = 0, \tag{5.20}$$

$$\frac{\partial S_0'}{\partial t} + \overline{v} \cdot \nabla_x S_0' + v' \cdot \nabla_x \overline{S_0} + v' \cdot \nabla_x S_0' - \overline{v' \cdot \nabla_x S_0'} + v \cdot \nabla_y w = 0.$$

We remark that the term $\overline{v' \cdot \nabla_x S_0'}$ in (5.20) plays a role similar to the Reynolds stress term in turbulence modeling. This is the term that introduces the nonlocal memory effect into the average equation.

The above multiscale analysis has been applied to upscale the saturation in the two-phase flow in [144]. To solve the coupled elliptic equation for pressure and the transport equation for saturation, we can use the IMPES method, where the pressure equation is solved using MsFVEM and then the velocity approximation is used for upscaling of the transport equation. In [144], the authors presented many numerical experiments for the immiscible flows in porous media based on a multiscale analysis similar to the one described here. They showed that their upscaling method captures both the average and the small-scale fluctuation very well for permeability fields described using two-point correlation functions. By using a new reparameterization technique introduced in [151], we have applied this upscaling method to simulate more realistic heterogeneous porous media without scale separation or periodic structure in [144].

5.2.9 Summary

In summary, the main purpose of this section has been to introduce a new (adaptive) multiscale method for solving the transport equation that arises in immiscible two-phase flow in porous media. The basic idea is to compute the global flow on a coarse grid, and map the averaged grid block saturations onto plausible saturation profiles on a finer subgrid. To enhance the accuracy of the coarse-grid saturation profile, while at the same time avoiding an upscaling phase involving, for example, the construction of pseudo-relative permeability functions, we introduce a numerical scheme for solving the transport equation on a coarse grid that honors fine-scale structures in the velocity field in a mathematically consistent manner. Moreover, to capture rapid transitions in saturation values near propagating saturation fronts accurately, we propose to include an adaptive component in the algorithm. In the adaptive algorithm, we solve the saturation locally on a fine grid in transient flow regions. The proposed (adaptive) multiscale method has been analyzed and tested on models with complex heterogeneous structures. We have also extended and implemented multiscale methods for transport equations on unstructured corner-point grids (see [6]). In this section, we also discussed a few other approaches for coarsening transport equations.

5.3 Applications to Richards' equation

5.3.1 Problem statement

In this section we consider the applications of MsFEMs to Richards' equation ([236]), which describes the infiltration of water flow into porous media whose pore space is filled with air and some water. The equation describing Richards' equation under some assumptions is given by

$$\frac{\partial}{\partial t}\theta(p) - \mathrm{div}(k(x,p)\nabla(p + x_3)) = 0 \ \text{ in } \Omega, \tag{5.21}$$

where $\theta(p)$ is the volumetric water content and p is the pressure. The following are assumed ([236]) for (5.21): (1) the porous media and water are incompressible; (2) the temporal variation of the water saturation is significantly larger than the temporal variation of the water pressure; (3) the air phase is infinitely mobile so that the air pressure remains constant (in this case it is atmospheric pressure which equals zero); and (4) neglect the source/sink terms.

Constitutive relations between θ and p and between $k(x,p)$ and p are developed appropriately, which consequently gives nonlinearity behavior in (5.21). The relation between the water content and pressure is referred to as the moisture retention function. The equation written in (5.21) is called the coupled-form of Richards' equation. This equation is also called the mixed form of Richards' equation, due to the fact that there are two variables involved in it, namely, the water content θ and the pressure head p. Taking advantage of the differentiability of the soil retention function, one may rewrite (5.21) as follows

$$C(p)\frac{\partial}{\partial t}p - \operatorname{div}(k(x,p)\nabla(p+x_3)) = 0 \;\text{ in } \Omega, \tag{5.22}$$

where $C(p) = d\theta/dp$ is the specific moisture capacity. This version is referred to as the head-form (h-form) of Richards' equation. Another formulation of the Richards' equation is based on the water content θ,

$$\frac{\partial}{\partial t}\theta - \operatorname{div}(D(x,\theta)\nabla\theta) - \frac{\partial k}{\partial x_3} = 0 \;\text{ in } \Omega, \tag{5.23}$$

where $D(\theta) = k(\theta)/(d\theta/dp)$ defines the diffusivity. This form is called the θ-form of Richards' equation.

The sources of nonlinearity of Richards' equation come from the moisture retention and relative hydraulic conductivity functions, $\theta(p)$ and $k(x,p)$, respectively. Reliable approximations of these relations are in general tedious to develop and thus also challenging. Field measurements or laboratory experiments to gather the parameters are relatively expensive, and furthermore, even if one can come up with such relations from these works, they will be somehow limited to the particular cases under consideration.

Perhaps the most widely used empirical constitutive relations for the moisture content and hydraulic conductivity is due to the work of van Genuchten [131]. He proposed a method of determining the functional relation of relative hydraulic conductivity to the pressure head by using the field observation knowledge of the moisture retention. In turn, the procedure would require curve-fitting to the proposed moisture retention function with the experimental/observational data to establish certain parameters inherent to the resulting hydraulic conductivity model. There are several widely known formulations of the constitutive relations: the Haverkamp model

$$\theta(p) = \frac{\alpha\,(\theta_s - \theta_r)}{\alpha + |p|^{\beta}} + \theta_r, \quad k(x,p) = k_s(x)\frac{A}{A + |p|^{\gamma}};$$

van Genuchten model [131]

$$\theta(p) = \frac{\alpha\,(\theta_s - \theta_r)}{\left[1 + (\alpha|p|)^n\right]^m} + \theta_r, \quad k(x,p) = k_s(x)\frac{\left\{1 - (\alpha|p|)^{n-1}\left[1 + (\alpha|p|)^n\right]^{-m}\right\}^2}{\left[1 + (\alpha|p|)^n\right]^{m/2}};$$

exponential model [268]

$$\theta(p) = \theta_s\, e^{\beta p}, \quad k(x,p) = k_s(x)\, e^{\alpha p}.$$

5.3.2 MsFVEM for Richards' equations

The spatial field $k_s(x)$ in the above models is also known as the saturated hydraulic conductivity. It has been observed that the hydraulic conductivity has a broad range of values, which together with the functional forms presented above confirm the nonlinear behavior of the process. Furthermore, the water content and hydraulic conductivity approach zero as the pressure head goes to very large negative values. In other words, the Richards' equation has a tendency to degenerate in a very dry condition, that is conditions with a large negative pressure. Because we are interested in mass conservative schemes, finite volume formulation (3.13) of the global problem instead of finite element formulation is used. For (5.21), it is to find $p_h \in W_h$ such that

$$\int_{V_z} (\theta(\eta^{p_h}) - \theta^{n-1})\, dx - \Delta t \int_{\partial V_z} k(x, \eta^{p_h}) \nabla p_{r,h} \cdot n\, ds = 0, \quad \forall z \in Z_h^0, \quad (5.24)$$

where θ^{n-1} is the value of $\theta(\eta^{p_h})$ evaluated at time-step $n-1$, and $p_{r,h} \in \mathcal{P}_h$ is a function that satisfies the boundary value problem:

$$\begin{aligned} -\mathrm{div}(k(x, \eta^{p_h}) \nabla p_{r,h}) &= 0 \quad \text{in } K, \\ p_{r,h} &= p_h \quad \text{on } \partial K. \end{aligned}$$

Here V_z is the control volume surrounding the vertex $z \in Z_h^0$ and Z_h^0 is the collection of all vertices that do not belong to the Dirichlet boundary (see Section 3.2).

MsFEM (or MsFVEM) offers a great advantage when the nonlinearity and heterogeneity of $k(x, p)$ are separable; that is

$$k(x, p) = k_s(x)\, k_r(p). \quad (5.25)$$

In this case, as we discussed earlier, the local problems become linear and the corresponding $\mathcal{P}_h$ is a linear space; that is we may construct a set of basis functions $\{\phi_z\}_{z \in Z_h^0}$ (as before) such that they satisfy

$$\begin{aligned} -\mathrm{div}(k_s(x) \nabla \phi_z) &= 0 \quad \text{in } K, \\ \phi_z &= \phi_z^0 \quad \text{on } \partial K, \end{aligned}$$

where ϕ_z^0 is a piecewise linear function. We note that if p_h has a discontinuity or a sharp front region, then the multiscale basis functions need to be updated only in that region. The latter is similar to the use of MsFEM in two-phase flow applications. In this case the basis functions are only updated along the front. Now, we may formulate the finite-dimensional problem. We want to seek $p_{r,h} \in \mathcal{P}_h$ with $p_{r,h} = \sum_{z \in Z_h^0} p_z \phi_z$ such that

$$\int_{V_z} (\theta(\eta^{p_h}) - \theta^{n-1})\, dx - \Delta t \int_{\partial V_z} k_s(x)\, k_r(\eta^{p_h})\, \nabla p_{r,h} \cdot n\, ds = 0,$$

for every control volume $V_z \subset \Omega$. To this equation we can directly apply the linearization procedure described in [133]. Let us denote

$$r^m = p_{r,h}^m - p_{r,h}^{m-1}, \quad m = 1, 2, 3, ..., \quad (5.26)$$

where $p^m_{r,h}$ is the iterate of $p_{r,h}$ at the iteration level m. Thus, we would like to find $r^m = \sum_{z \in Z^0_h} r^m_z \phi_z$ such that for $m = 1, 2, 3, ...$ $\|r^m\| \leq \delta$ with δ being some pre-determined error tolerance

$$\int_{V_z} C(\eta^{p_h^{m-1}})\, r^m\, dx - \Delta t \int_{\partial V_z} k_s(x)\, k_r(\eta^{p_h^{m-1}})\, \nabla r^m \cdot n\, ds = R^{h,m-1},$$

with

$$R^{h,m-1} = -\int_{V_z} (\theta(\eta^{p_h^{m-1}}) - \theta^{n-1})\, dx + \Delta t \int_{\partial V_z} k_s(x)\, k_r(\eta^{p_h^{m-1}})\, \nabla p^{m-1}_{r,h} \cdot n\, ds. \tag{5.27}$$

The superscript m at each of the functions means that the corresponding functions are evaluated at an iteration level m.

5.3.3 Numerical results

We present several numerical experiments that demonstrate the ability of the coarse models presented in the previous subsections. The coarse models are compared with the fine model solved on a fine mesh. We have employed a finite volume difference to solve the fine-scale equations. This solution serves as a reference for the proposed coarse models. The problems that we consider are typical water infiltration into an initially dry soil. The porous media that we consider is a rectangle of size $L_1 \times L_2$ (see Figure 5.8). The fine model uses 256×256 rectangular elements, and the coarse model uses 32×32 rectangular elements.

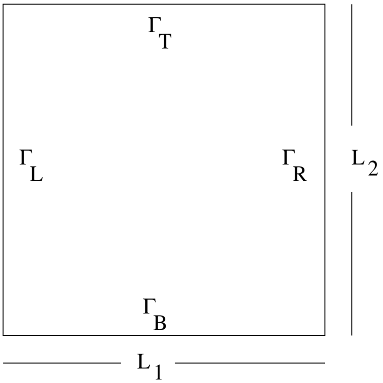


Fig. 5.8. Rectangular layout of porous media.

A realization of the hydraulic conductivity field $k_s(x)$ is generated using geostatistical package GSLIB ([85]). We have used a spherical variogram with prescribed correlation lengths (l_1, l_2) and the variance (σ) for this purpose. All examples use $\sigma = 1.5$.

The first problem is a soil infiltration, which was first analyzed by Haverkamp (cf. [64]). The porous media dimension is $L_1 = 40$ and $L_2 = 40$. The boundary conditions are as follows. Γ_L and Γ_R are impermeable, and Dirichlet conditions are imposed on Γ_B and Γ_T, namely $p_T = -21.7$ in Γ_T, and $p_B = -61.5$ in Γ_B. The initial pressure is $p_0 = -61.5$. We use the Haverkamp model to construct the constitutive relations. The related parameters are $\alpha = 1.611 \times 10^6$, $\theta_s = 0.287$, $\theta_r = 0.075$, $\beta = 3.96$, $A = 1.175 \times 10^6$, and $\gamma = 4.74$. For this problem we assume that the nonlinearity and heterogeneity are separable, where the latter comes from $k_s(x)$ with $\overline{k_s} = 0.00944$. We assume that appropriate units for these parameters hold. There are two cases that we consider for this problem, namely the isotropic heterogeneity with $l_1 = l_2 = 0.1$, and the anisotropic heterogeneity with $l_1 = 0.01$ and $l_2 = 0.20$. For the backward Euler scheme, we use $\Delta t = 10$. Note that the large value of Δt is due to the smallness of $\overline{k_s}$ (average magnitude of the diffusion). The comparison is shown in Figures 5.9 and 5.10, where the solutions are plotted at $t = 360$.

The second problem is a soil infiltration through porous media whose dimension is $L_1 = 1$ and $L_2 = 1$. The boundary conditions are as follows. Γ_L and Γ_R are impermeable. Dirichlet conditions are imposed on Γ_B with $p_B = -10$. The boundary Γ_T is divided into three parts. On the middle part, a zero Dirichlet condition is imposed, and the rest are impermeable. We use the exponential model to construct the constitutive relations. with the following related parameters: $\beta = 0.01$, $\theta_s = 1$, $\overline{k_s} = 1$, and $\overline{\alpha} = 0.01$. The heterogeneity comes from $k_s(x)$ and $\alpha(x)$. Clearly, for this problem the nonlinearity and heterogeneity are not separable. Again, isotropic and anisotropic heterogeneities are considered with $l_1 = l_2 = 0.1$ and $l_1 = 0.20$, $l_2 = 0.01$, respectively. For the backward Euler scheme, we use $\Delta t = 2$. The comparison is shown in Figures 5.11 and 5.12, where the solutions are plotted at $t = 10$.

We note that the problems that we have considered are vertical infiltration on the porous media. Hence, it is also useful to compare the cross-sectional vertical velocity that will be plotted against the depth z. Here, the cross-sectional vertical velocity is obtained by taking an average over the horizontal direction (x-axis).

Figure 5.13 shows comparison of the cross-sectional vertical velocity for the Haverkamp model. The average is taken over the entire horizontal span because the boundary condition on Γ_T (and also on Γ_B) is all Dirichlet condition. Both plots in this figure show a close agreement between the fine and coarse models. For the exponential model, as we have described above, there are three different segments for the boundary condition on Γ_T; that is a Neumann condition on the first and third part, and a Dirichlet condition on the second/middle part of Γ_T. Thus, we compare the cross-sectional vertical velocity in each of these segments separately. Figure 5.14 shows the comparison for one of these segments. The agreement between the coarse-grid and fine-grid calculations is excellent.

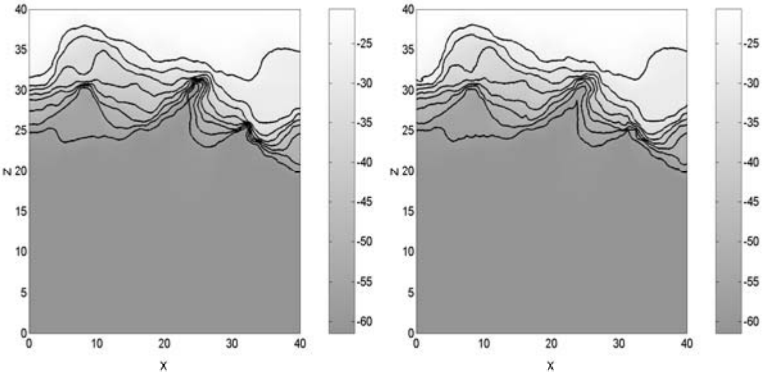


Fig. 5.9. Haverkamp model with isotropic heterogeneity. Comparison of water pressure between the fine model (left) and the coarse model (right).

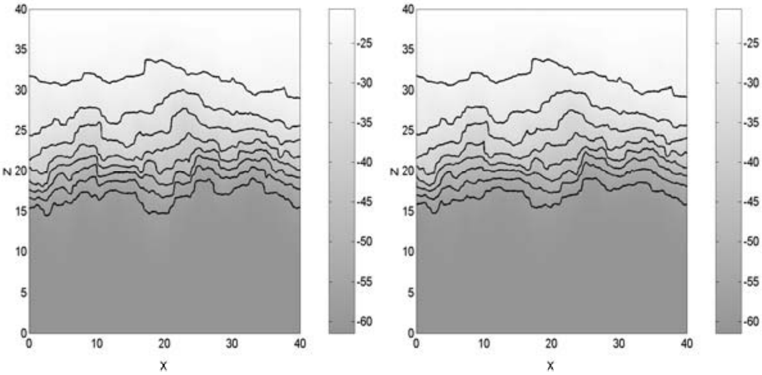


Fig. 5.10. Haverkamp model with anisotropic heterogeneity. Comparison of water pressure between the fine model (left) and the coarse model (right).

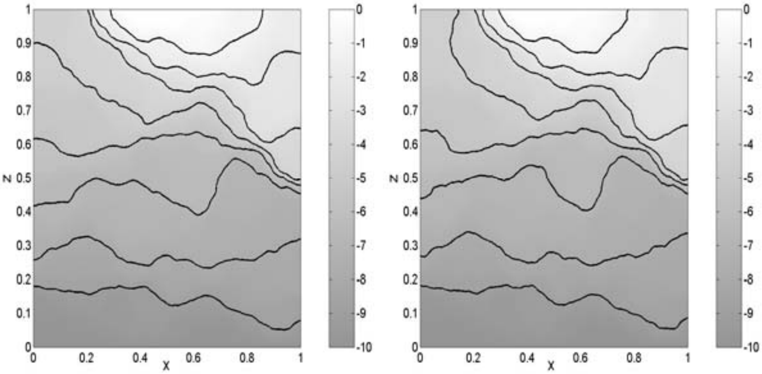


Fig. 5.11. Exponential model with isotropic heterogeneity. Comparison of water pressure between the fine model (left) and the coarse model (right).

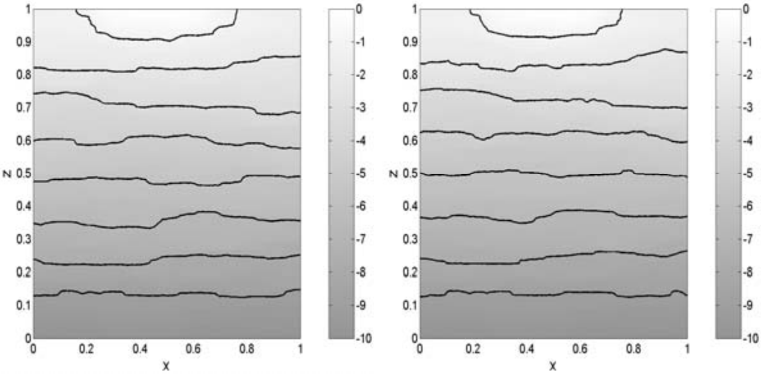


Fig. 5.12. Exponential model with anisotropic heterogeneity. Comparison of water pressure between the fine model (left) and the coarse model (right).

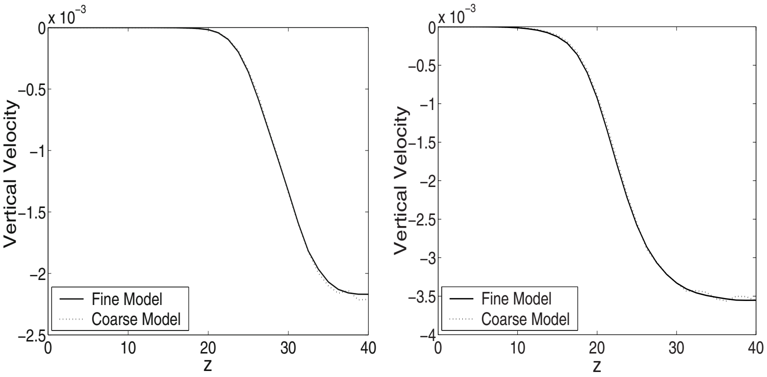


Fig. 5.13. Comparison of vertical velocity on the coarse grid for Haverkamp model: isotropic heterogeneity (left) and anisotropic heterogeneity (right).

5.3.4 Summary

In summary, the main goal of this section has been to apply MsFEMs to Richards' equations described by nonlinear parabolic equations. In particular, the MsFVEM for nonlinear problems developed in Section 3.2 is used for solving Richards' equation on the coarse grid. We presented numerical results for various heterogeneous hydraulic conductivity fields. Our numerical results show that MsFEMs can be used with success in predicting the solution on the coarse grid.

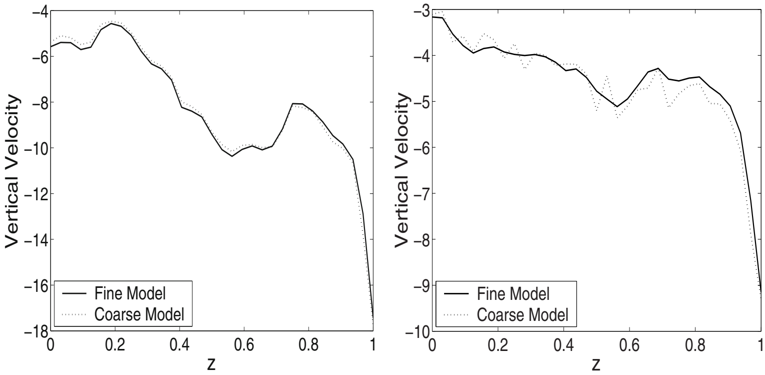


Fig. 5.14. Comparison of vertical velocity on the coarse grid for exponential model: isotropic heterogeneity (left) and anisotropic heterogeneity (right). The average is taken over the second third of the domain.

5.4 Applications to fluid–structure interaction

5.4.1 Problem statement

MsFEMs can also be used to solve complex multiphysics problems. In this section, we extend the MsFEM to solving a fluid–structure interaction (FSI) problem on the coarse grid. At the fine scale, we consider Stokes flow past an elastic skeleton. Thus, our domain Ω has two parts: a fluid domain Ω_0^f and a solid domain Ω_0^s. The subscript 0 indicates that these are the domains of the two constituents (solid and fluid) at rest. As macroscopic boundary conditions are applied, the fluid starts to flow, thus exerting forces on the solid, causing them to deform. As a steady state is achieved the fluid flows in a domain $\Omega^f = \Omega \setminus \Omega^s$, and the forces that the fluid exerts on the solid at their interface $\Gamma = \left(\partial\Omega^f \cap \partial\Omega^s\right) \setminus \partial\Omega$ are balanced by the elastic stresses inside the solid. The precise formulation of the FSI problem is:

$$\Gamma = \{X + u(X) | \forall X \in \Gamma_0\}, \tag{5.28}$$

$$-\mu\Delta v + \nabla p = b \text{ in } \Omega^f, \quad \text{div}(v) = 0 \text{ in } \Omega^f, \quad v = 0 \text{ on } \Gamma, \tag{5.29}$$

$$-\text{div}(S(E)) = b_0 \text{ in } \Omega_0^s, \tag{5.30}$$

$$\det(\nabla u + I)(-pI + 2\mu D(x(X)))\left(\nabla u + I\right)^{-T} n_0 = S(E)n_0 \text{ on } \Gamma_0. \tag{5.31}$$

Note that Γ is the set of points $X + u(X)$. The above equation utilizes the standard notation from continuum mechanics. The deformation gradient $F(X) = \nabla x(X)$, the displacements in the solid $u(X) = x(X) - X$, the infinitesimal strain $E(X) = \frac{1}{2}\left(\nabla u(X) + \nabla u(X)^T\right)$, the fluid velocity $v(x)$ and, finally, the symmetric part of its gradient $D(x) = \frac{1}{2}\left(\nabla v(x) + \nabla v(x)^T\right)$. Furthermore, the usual Cauchy stress tensor is denoted by $T(x)$, which is the

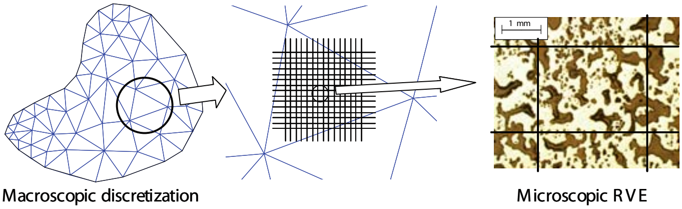


Fig. 5.15. Schematic of nonlinear MsFEM for FSI. The microstructure shown is an actual sample of porous shape memory alloy [172].

convenient stress measure when describing the fluid. Observe that Cauchy stress tensor is a spatial field, defined on the deformed configuration of the body. For the solid part, the first Piola-Kirchhoff stress tensor $S(X)$ is more appropriate as it gives a description of the stresses in Lagrangian coordinates. The two are related by the identity (see, e.g., [157]):

$$S(X) = \det(F(X))T(x(X))F^{-T}(X). \tag{5.32}$$

In the above formulation of the FSI system the constitutive equation for the Piola-Kirchoff stress S is left unspecified. It has to be specified taking into account the particular solid at hand (see [157] for details). In our numerical examples, the linear elasticity model will be used; that is $S(E) = \mathcal{C} : E$. We refer, for example, to [158, 232] for full details on deriving the FSI problem.

Observe that the position of the interface is a part of the boundary value problem, and the solid–fluid coupling term (5.31) is nonlinear in u. Therefore, the FSI problem is nonlinear, even when the constitutive equation for the solid is a linear one.

5.4.2 Multiscale numerical formulation

The mapping E^{MsFEM}, which couples the coarse scale pressure p_0 and displacements u_0 to the fine-scale fluid velocity v, pressure p and displacements u is defined through the fine-scale FSI problem (5.28)-(5.31). In our problem, we use RVE for local computations (see Figure 5.15 for the illustration[1]). Note that E^{MsFEM} defines a map from a coarse-scale solution $\{p_h, u_h\}$ with given $\mathit{\Gamma}_0$ to a fine-scale approximation $\{p_{r,h}, v_{r,h}, u_{r,h}\}$ via the local solution of (5.28)-(5.31). Various boundary conditions can be chosen for local problems. In our simulations, we use periodic boundary conditions such that the spatial averages of $p_{r,h}$ and $u_{r,h}$ are the same as those for p_h and u_h. In general, one can also take p_h and u_h as boundary conditions. In the computation of the local FSI solution with given p_h, u_h and the reference interface $\mathit{\Gamma}_0$, one solves an

[1] The right figure is the courtesy of the Shape Memory Alloy Research Team (SMART) at Texas A & M University

iterative problem. We assume that this iterative problem converges and provides a unique fine-scale solution $\{p_{r,h}, v_{r,h}, u_{r,h}\}$. This condition guarantees that E^{MsFEM} is a single-valued map.

Next, we discuss the coarse-scale formulation of the problem. In coarse-scale simulations, our goal is to find approximations of p_0 and u_0, denoted by p_h and u_h. When substituting $(p_{r,h}, v_{r,h}, u_{r,h})$ (given p_h and u_h) into the fine-scale equations, one needs to solve the resulting system on the coarse-dimensional space. There are various approaches as discussed earlier (see Section 2.4). In particular, one can multiply the fine-scale residual by coarse-scale test functions, or minimize the residual at some coarse points, or use coarse-scale equations when available. These procedures result in a nonlinear equation for finding (p_h, u_h)

$$G(p_h, u_h) = 0, \tag{5.33}$$

where G is the reduced variational formulation. This equation is solved via a fixed-point iteration. Here, we consider a simple, physically intuitive, iterative method. In particular, we assume that the coarse-scale equation for the pressure is given by the Darcy equation (see (1.1)) and the coarse-scale equation for the elasticity has the same form as the underlying fine-scale equations, but with upscaled elastic properties that are computed based on local RVE computations. We carry out numerical simulations iteratively. Given p_h^n and u_h^n at the nth iteration, $p_{r,h}^n$, $v_{r,h}^n$, and $u_{r,h}^n$ are computed. This is done by using a local problem in RVE as described above. This step involves the solution of the elasticity problem and yields new pore geometry based on the deformations. Furthermore, taking into account local geometry of the pore space, the permeabilities k^n and upscaled elastic properties are computed via standard cell problems (e.g., [240, 42]). Once the permeabilities are computed, the global problem

$$\mathrm{div}(k^n(x)\nabla p^{n+1}) = f$$

is solved and p_h^{n+1} (finite element projection of p^{n+1}) is calculated. Similarly, u_h^{n+1} is computed by solving elasticity equation with upscaled elastic properties (e.g., $\mathcal{C}^*$ for linear solids). This iterative procedure can be summarized in Algorithm 5.4.1. Modifications of this algorithm are presented in [232].

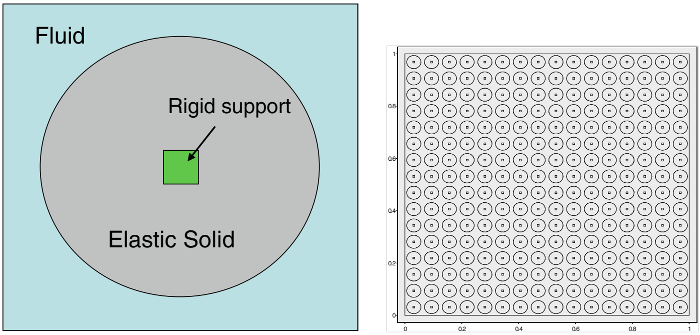
Algorithm 5.4.1 Iterative homogenization of strongly coupled FSI problem

- Initialize all micro- and macro-fields to zero.
- Project p_h^n and u_h^n using E^{MsFEM}.
- Evaluate the permeability and elastic properties in a coarse-grid block that involves the computation of the deformed pore geometry.
- Compute macroscopic quantities p_h^{n+1} and u_h^{n+1}.
- Check for convergence and, if necessary, return to Step 2.

In the numerical examples, we consider flow past elastic obstacles. The fluid surrounds the obstacles and the obstacles are supported rigidly in their center. Note that the rigid support is necessary, otherwise the flow will move them. Observe also that in 2D either the fluid or the solid domains can be connected, but not both. Therefore, to study upscaling of deformable porous media the solid domain has to be disconnected, so that the fluid can flow throughout the domain and interact with the solid. This simplification allows us to formulate a coarse-scale equation for the macroscopic pressure p_0 only, and thus, E^{MsFEM} is defined for a given p_h.

5.4.3 Numerical examples

In the numerical examples, we consider flow past a 2D periodic arrangement of elastic obstacles (Figure 5.16). The obstacles are, in the reference configuration, circular and centered in the middle of a square unit cell (Figure 5.16(a)). The macroscopic domain is assumed periodic (with the period size ϵ) in the reference configuration and we consider a series of macroscopic domains with $\epsilon^{-1} = 4, 8, 16, ...$ The case $\epsilon = 1/16$ is shown in Figure 5.16(b).



(a) Unit cell

(b) Macroscopic domain

Fig. 5.16. The unit cell (a) consists of circular linear elastic material, surrounded by the fluid. The elastic media is supported rigidly in the center. The unit cell is arranged periodically to form the macroscopic domain. (b) Macroscopic domain with a 16 × 16 periodic arrangement of the unit cell (a).

The elastic material under consideration is linear and isotropic with Young's modulus $E = 1.44$ and Poisson's ratio $\nu = 0.1$. The fluid has viscosity 0.1. These non-dimensional properties are selected such that a pressure in the range $0.1 - 0.5$ will produce a sizeable deformation in the solid and lead to strongly coupled FSI problems.

The main objectives of this example are to demonstrate the behavior of the iterative Algorithm 5.4.1. There are two main questions that need to be illuminated: first, whether the nonlinear iteration converges, and second, the approximation that Algorithm 5.4.1 provides to the fine-scale solution of the FSI problem needs to be investigated with respect to the scale parameter ϵ.

The boundary value problem is thus designed to meet both of these goals. The macroscopic domain is the unit square (Figure 5.16(b)) and a uniform pressure P_l is applied at the left side of the domain. The pressure at the right side is 0 and no-flow boundary conditions are considered at the top and bottom sides of the domain. These boundary conditions imply the fine-scale solution is periodic in the x_2-direction with the period being one horizontal strip of $(1/\epsilon)$ unit cells. Also, the averaged macroscopic quantities are essentially one-dimensional. This very simple boundary value problem is selected to allow direct numerical simulations (DNS) of the fine-scale solution to the FSI problem. A DNS is computationally very intensive both in memory consumption and CPU time. However, with the selected boundary conditions, a DNS can be performed on a single strip of unit cells and then periodically repeated in the x_2-direction. This leads to a factor of $1/\epsilon$ reduction in computational effort and allows us to compute the DNS solution on a series of domains with $\epsilon^{-1} = 4, 8, 16, 32, 64$.

We perform a series of computations with $P_l = 0.1$ and $P_l = 0.2$. The first observation is that Algorithm 5.4.1 in fact behaves as a contraction operator and converges. The approximate upscaled pressure is plotted in Figure 5.17. Because, as already discussed, the upscaled pressure does not vary in the x_2 due the boundary conditions, the plot is a cross-section of the upscaled pressure at a fixed location $x_2 = \text{const}$. The number of iterations it took for Algorithm 5.4.1 to reach a relative accuracy of $\times 10^{-6}$ is reported in Table 5.1.

Based on the results it is seen that the algorithm behaves as a contraction

Table 5.1. Performance of Algorithm 5.4.1. Listed are the iteration number it took Algorithm 5.4.1 to converge as well as the error between the ”exact” DNS and the MsFEM (fine-scale) solution for fine-scale displacements.

ϵ	$P_l = 0.1$				
	Iterations	L^∞ Error	L^∞ Rel. Error	L^2 Error	L^2 Rel. Error
1/4	6	1.23×10^{-3}	0.18	2.48×10^{-4}	0.23
1/8	6	3.18×10^{-4}	0.10	4.39×10^{-5}	0.13
1/16	6	8.07×10^{-5}	0.053	7.75×10^{-6}	0.069
1/32	6	2.03×10^{-5}	0.027	1.37×10^{-6}	0.0351
	$P_l = 0.2$				
	Iterations	L^∞ Error	L^∞ Rel. Error	L^2 Error	L^2 Rel. Error
1/4	8	2.96×10^{-3}	0.22	4.93×10^{-4}	0.22
1/8	8	7.94×10^{-4}	0.126	8.78×10^{-5}	0.127
1/16	8	2.06×10^{-4}	0.068	1.56×10^{-5}	0.067
1/32	8	5.25×10^{-5}	0.035	2.75×10^{-6}	0.034

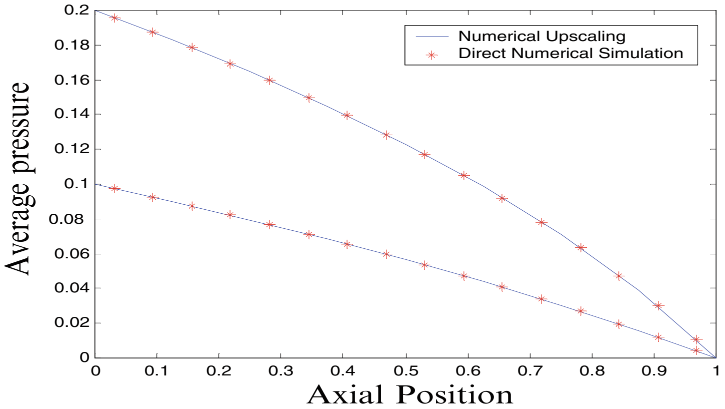


Fig. 5.17. Comparison of coarse-scale pressure profiles at $x_2 = 0.5$.

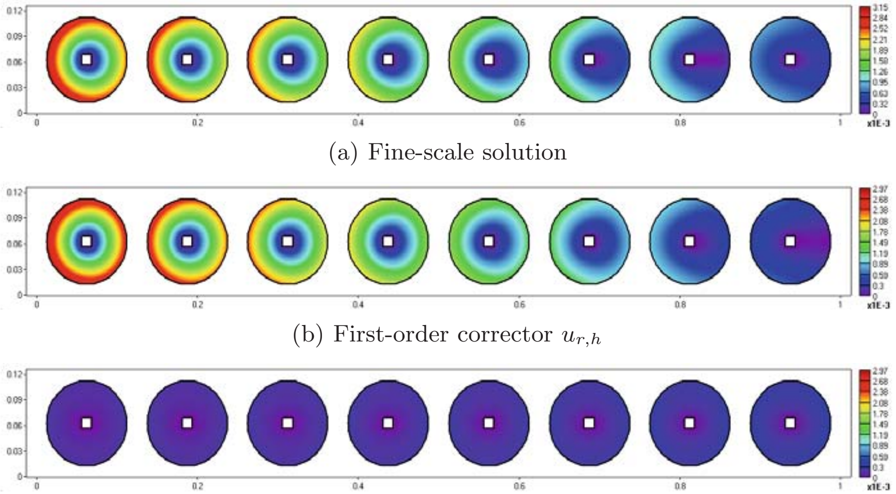
operator and the number of iterations is independent of the length scale ϵ. The same table also lists the comparisons of the DNS solution with the projected fine-scale displacements (via the mapping E^{MsFEM}). Based on the error between the DNS displacements and the fine-scale displacements obtained via the MsFEM, it is seen that the method is convergent in terms of ϵ. The actual convergence rate requires a detailed theoretical analysis which is reported in [232]. [232].

5.4.4 Discussions

In this section, the application of nonlinear MsFEM to complex multiphysics problems was studied. Here, our goal was simply to discuss an application of the MsFEM to FSI problems and we did not discuss many other existing methods (e.g., [141, 123, 123, 174]). Note that the governing equations do not have elliptic or parabolic forms such as those discussed in Chapter 3, but the general concept of MsFEMs (see Section 2.4) can be applied for solving such systems.

5.5 Applications of mixed MsFEMs to reservoir modeling and simulation (by J. E. Aarnes)

Reservoir simulation — the modeling of flow and transport of hydrocarbons in oil and gas reservoirs — is perhaps the most widely considered application in the literature on numerical models for porous media flow. In fact, numerical reservoir simulation has a history that goes back to the early days of the computer. Due to constraints on computational capability, reservoir simulation has been performed on very coarse models with limited spatial



(a) Fine-scale solution

(b) First-order corrector $u_{r,h}$

(c) Error between fine-scale solution and $u_{r,h}$

Fig. 5.18. Displacements in a typical solution to the model problem. The macroscopic domain has 8×8 unit cells, and due to periodicity in the x_2-direction, only one horizontal row of unit cells is shown. The exact fine-scale displacements (a) can be compared with the first-order corrector (b). The difference between the two is shown in (c).

resolution. However, the current trend in geomodeling – the process of developing a conceptual geological description of the reservoir – is to build detailed high-resolution models that match as closely as possible the geologists' perception of the reservoir. As a result, there is a steadily increasing gap between the size of the geological model built by geologists, and the model used for reservoir simulation. The reservoir simulation model is normally obtained by coarsening or upscaling the geological model.

As an alternative to upscaling it has been suggested that multiscale methods can be used to run simulations directly on geological models. To this end it is generally assumed that the fine and coarse grids overlap such that each block in the coarse grid simply consists of a number of cells from the underlying fine grid. This means that one can perform the coarsening of the grid in index space rather than in physical space, and thereby significantly simplify the process of generating the coarse grid. In particular, one avoids the practical problems of resampling nonoverlapping cells/blocks in the fine/coarse grid that are traditionally associated with upscaling.

In this section we make an effort to demonstrate some applications where multiscale flow solvers used in combination with various methods for fast computation of fluid transport may spur new ways of using flow information as part of reservoir planning and management. In particular, we demonstrate how the multiscale methods can be used to

- Accelerate the solution of the pressure equation in three-phase black oil reservoir simulation models (and retain the solution accuracy).
- Provide very rapid estimation of production characteristics on flow-grids that are tuned to reservoir flow patterns.
- Almost instantly estimate injector–producer pairs and swept volumes.

The simulations were performed in Matlab on a desktop computer with a dual AMD Athlon X2 4400+ processor with 1 MB cache and 2 GB memory.

5.5.1 Multiscale method for the three-phase black oil model

The mixed MsFEM discussed before, which was first introduced by Chen and Hou [71], has later been modified in a sequence of papers [1, 11, 13, 12] to handle the geometric and physical complexity of real-field reservoir models. For instance, whereas the original method was developed for solving elliptic problems on Cartesian grids, the most recent version [12] is designed for solving the parabolic pressure equation of three-phase black oil models on real-field corner-point grids with faults. The three-phase black oil model describes the flow of an aqueous phase (a), usually water, a liquid phase (l) containing oil and liquefied gas, and a vapor phase (v) containing gas and vaporized oil. The pressure equation for the three-phase black oil model may be expressed on the following form:

$$\left(\frac{\partial \phi_{\text{por}}}{dp_l} + \phi_{\text{por}} \sum_j c_j S_j\right) \frac{\partial p_l}{dt} + \nabla \cdot \left(\sum_j v_j\right) + \sum_j c_j v_j \cdot \nabla p_l = q, \quad (5.34)$$

where p_l is liquid pressure, ϕ_{por} is porosity, v_j, c_j and S_j are phase velocities, compressibilities, and saturations, respectively, and q is a volumetric source term. The phase velocities are related to the phase pressures p_j through Darcy's law:

$$v_j = -\frac{k k_{rj}}{\mu_j} \left(\nabla p_j + \rho_j g e_3\right), \quad j = a, l, v. \quad (5.35)$$

Here ρ_j is the density of phase j, g is the magnitude of acceleration of gravity, e_3 is the unit normal pointing vertically upwards, k is the absolute permeability, and k_{rj} and μ_j are the relative permeability and viscosity of phase j, respectively. See [12] for the definition of the phase compressibilities.

When applied to the three-phase black oil model the mixed MsFEM approximates the liquid pressure p_l and the total velocity $v = \sum_j v_j$ in finite-dimensional subspaces defined over the coarse grid. Recall that the pressure is approximated in a regular mixed finite-element space consisting of functions that are constant on each coarse block, and the velocity is approximated in a special multiscale space spanned by special multiscale basis functions that correspond to localized solutions of the pressure equation with a prescribed direction of flow; see [1, 13]. Given these basis functions, the mixed MsFEM finds the best linear superposition (in a certain sense) under the constraint that the velocity field is mass conservative on the coarse scale. Moreover, if the local flow solutions are mass conservative on the fine grid, then so will the global mixed MsFEM solution be.

To perform a reservoir simulation using the mixed MsFEM one proceeds as follows.

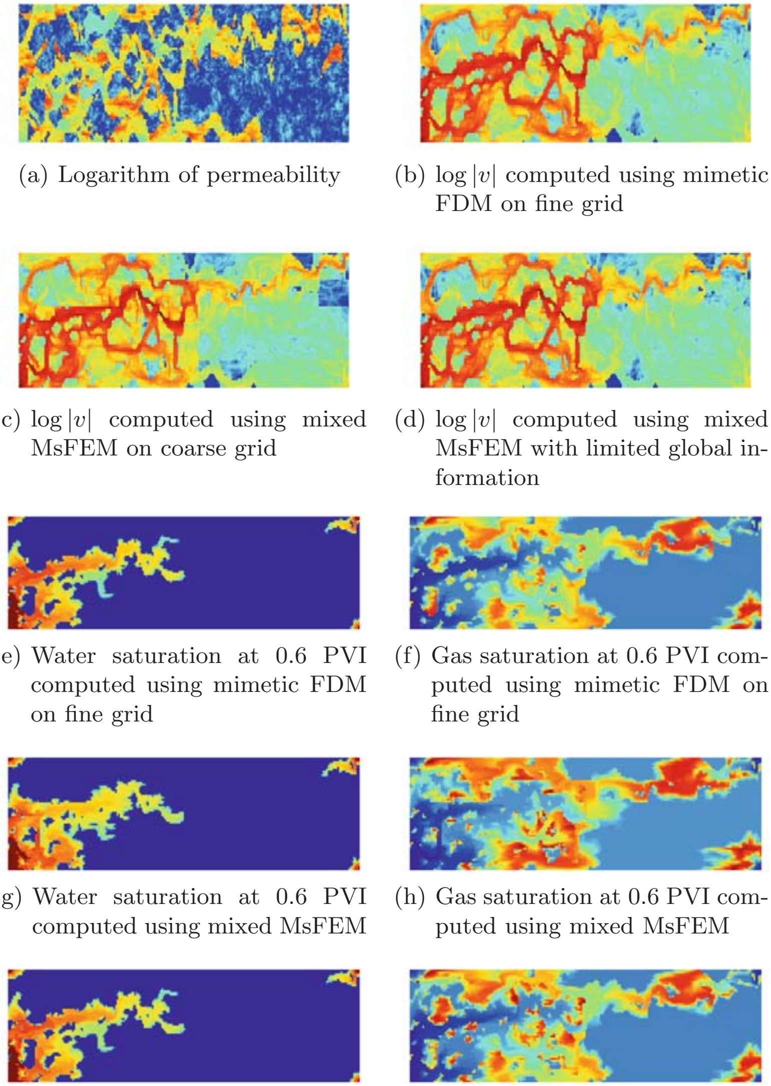
1. Introduce a coarse grid, for instance, by partitioning in index space as seen in Figure 2.10.
2. Detect all pairs of adjacent blocks.
3. For each pair, compute a velocity basis function.
4. Start simulation, for each time-step, do
   a) Assemble and solve the coarse-grid system.
   b) Recover fine-grid velocities/fluxes.
   c) Solve the fluid-transport equations.

For increased stability, one may iterate on solving the pressure and transport equations before advancing to the next time-step and thereby obtain a fully implicit method [183]. Similarly, for cases with strong displacement fronts, one may also update a few basis functions throughout the simulation to account more accurately for a strong saturation dependence; see [1, 167].

To illustrate the accuracy of the multiscale solutions, we consider a two-dimensional test-case modeling layer 68 from model 2 of the SPE comparative solution project [78], henceforth called the SPE 10 model. This particular layer is known to be a very difficult model, (see, e.g., [167]). The simulations start with 0.4 PVI of gas injection followed by 0.6 PVI of water injection. The reservoir is initially filled with 5% gas and 95% oil, four injection wells constrained to inject at 300 bar are located at each corner, and one rate-constrained production well is located in the middle.

We consider both the mixed MsFEM in [13] and the corresponding method using limited global information to define the multiscale basis functions. The fine grid is a 60-by-220 Cartesian grid and the coarse grid for the mixed MsFEM is defined to be a 5-by-11 Cartesian grid. Accuracy of the mixed MsFEM solutions is assessed by comparing the water-cut (fraction of water in produced fluid) and gas-cut (fraction of gas in produced fluid) curves obtained using a mixed MsFEM with the corresponding curves obtained by solving the pressure equation on the fine grid using a mimetic finite difference method (FDM) [177, 41]. The latter solution is referred to as the reference solution. In all simulations the saturation equations are solved on the fine grid.

Figure 5.19 shows the logarithm of permeability, magnitude of velocity at initial time, and water-cut and gas-cut as functions of PVI (pore volume injected). Although there are certain differences between $\log |v|$ computed using a mixed MsFEM (without limited global information) and the fine-grid solution depicted in Figure 5.19(b), we see that the mismatch has limited influence on the production curves. Indeed, even with a coarse grid with only 55 blocks the mixed MsFEM produces water-cut and gas-cut curves that match the reference solution closely. This demonstrates that the mixed MsFEM captures the main flow characteristics.



(a) Logarithm of permeability

(b) $\log|v|$ computed using mimetic FDM on fine grid

(c) $\log|v|$ computed using mixed MsFEM on coarse grid

(d) $\log|v|$ computed using mixed MsFEM with limited global information

(e) Water saturation at 0.6 PVI computed using mimetic FDM on fine grid

(f) Gas saturation at 0.6 PVI computed using mimetic FDM on fine grid

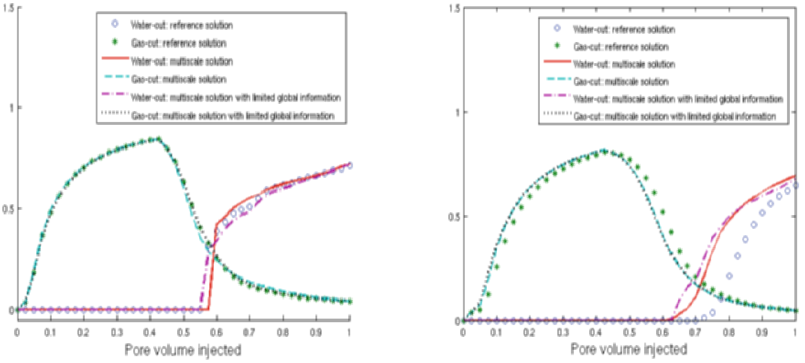
(g) Water saturation at 0.6 PVI computed using mixed MsFEM

(h) Gas saturation at 0.6 PVI computed using mixed MsFEM

(i) Water saturation at 0.6 PVI computed using mixed MsFEM with limited global information

(j) Gas saturation at 0.6 PVI computed using mixed MsFEM with limited global information

Fig. 5.19. Velocity solutions at initial time and saturation profiles at 0.6 PVI.



(a) Simulations on layer 68 from the SPE10 model

(b) Simulations on layers 81–85 from the SPE10 model

Fig. 5.20. Water-cut and gas-cut for simulations on a two-dimensional model (layer 68 from SPE 10) and a three-dimensional model (layers 81–85 from SPE10).

When solving the pressure equation on the fine grid using the mimetic FDM the time spent on solving the pressure equation stands for 86% of the computation time. In the multiscale simulations, on the other hand, the time spent on solving the saturation equations dominates the computation time (54%). With our current Matlab implementation the multiscale simulations run seven times faster than the fine-grid simulation. On larger models the difference will generally be more substantial because the computational complexity of the mixed MsFEM scales linearly with the model size. Moreover, the mixed MsFEM is very easily parallelized: the assembly of the coarse-grid system, which accounts for nearly 100% of the computation time, is called embarrassingly parallel, and perfectly suited for the multicore computers and distributed memory computing platforms. A further reduction in the computation time spent on solving the pressure equation, or alternatively an increase in model size, can therefore easily be achieved with parallel computing resources. However, to reduce the total computation time further one should also consider alternative strategies for solving the saturation equations.

Our numerical results show that mixed MsFEMs using limited global information give two-fold improvement in water and gas saturation errors when single-phase flow information is used in the construction of multiscale basis functions as discussed in Section 4.2. These results will be reported elsewhere.

5.5.2 Adaptive coarsening of the saturation equations

For large problems solving the saturation equations on the fine grid with a finite difference method may not be feasible, or may become a bottleneck. An alternative is to employ streamline methods [83] that advect the fluid phases along one-dimensional trajectories tangential to the velocity. These methods are generally very fast provided large time-steps between each pressure step can be taken. But it is also natural to ask if it is possible to exploit fine-grid velocity resolution in a multiscale type approach for the saturation equation. However, modeling the flow and transport accurately on coarse grids is difficult due to the dynamic nature of coarse-grid relative permeability functions [37] and the need to capture sharp propagating fronts. Fortunately, recent work [9] shows that one can model the main flow characteristics on relatively coarse grids without using pseudo-functions provided the coarse grid adapts to the local heterogeneity and resolves the dominant features in the velocity field (e.g., high-flow channels). In the following, we present an approach for generating such coarse grids and demonstrate how these grids can be used to get accurate production data.

Assume that the velocity is modeled on a high-resolution model (e.g., using the subresolution in mixed MsFEMs), and that it is prohibitively computationally expensive to solve the saturation equations on the same grid. Thus, we propose creating an upscaled model only for the saturation equation. This is done by generating a coarse grid that resolves underlying flow patterns more accurately than traditional coarse grids used in reservoir simulation. These grids allow us to capture more accurately flow quantities of interest, such as production characteristics, without resorting to multiphase upscaling.

As for a mixed MsFEM, we use the term *block* to denote a cell in the coarse grid to distinguish it from a cell in the fine grid. The coarsening strategy presented in [9], henceforth called the nonuniform coarsening algorithm, is essentially based on grouping cells according to flow magnitude. The algorithm involves two parameters that determine the degree of coarsening: a lower bound $V_{\min}$ on the volume of each block and an upper bound $G_{\max}$ on total amount of flow through each block. These parameters are selected to give the desired resolution of the saturation. A general rule for how to select the parameters is given in [9].

The steps in the nonuniform coarsening algorithm are as follows:

1. Use the logarithm of the velocity magnitude in each cell to segment the cells in the fine grid into ten different bins; that is, each cell c is assigned a number $n(c) = 1, \dots, 10$ by upper-integer interpolation in the range of $g(c) = \frac{10(\log|v(c)| - \min \log|v|)}{\max \log|v| - \min \log|v|}$.
2. Create an initial coarse grid with one block assigned to each connected collection of cells with the same value of $n(c)$.
3. Merge each block with less volume than $V_{\min}$ with a neighboring block.
4. Refine each block that has more flow than $G_{\max}$.
5. Repeat Step 3 and terminate.

Note that only the saturation equations are discretized on this grid. To this end, we employ a backward Euler method where the spatial discretization is a finite volume method that is upstream weighted at the fine-grid level; see [9].

This implies that we utilize the fine-grid resolution in the velocity field when solving for saturation on the coarse grid.

Figures 5.21(b)–5.21(d) show the logarithm of a velocity field as a piecewise constant function on the fine grid, on a coarse Cartesian grid with 240 blocks, and on a nonuniformly coarsened grid with 236 blocks. If we denote the reservoir by Ω, and N is the number of cells in the fine grid, then the nonuniform coarse grid is generated under the constraint that each block B satisfies

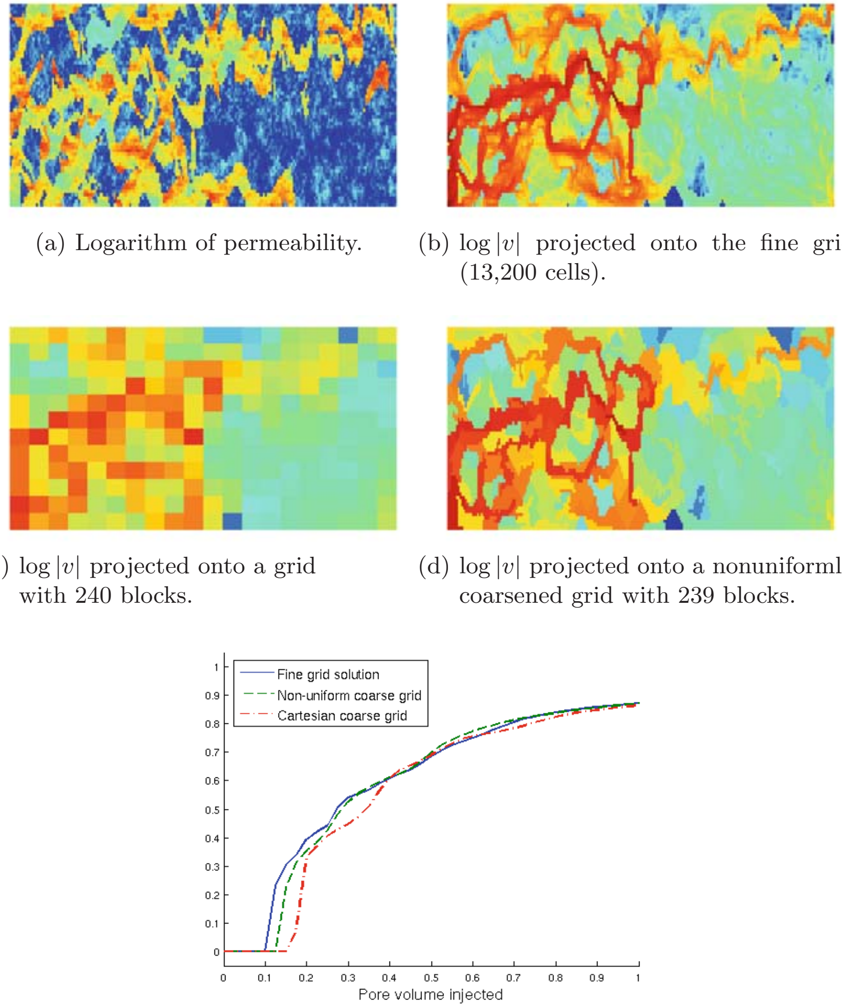
$$\int_B dx \geq \frac{15}{N}\int_\Omega dx \qquad \text{and} \qquad \int_B \log|v| dx \leq \frac{75}{N}\int_\Omega \log|v|\, dx.$$

We clearly see that the nonuniformly coarsened grid adapts to underlying flow patterns. In contrast, the channels with high velocity are almost impossible to detect in Figure 5.21(c). The fact that the coarse grid is capable of resolving the main flow trends leads to improved accuracy in modeled production characteristics. This is illustrated in Figure 5.21(e) which shows water-cut curves obtained on the nonuniform coarse grid are closer to the water-cut curve obtained on the fine grid than the water-cut curve obtained on the Cartesian coarse grid.

The robustness and accuracy of the nonuniform coarsening approach relative to modeling saturation on uniformly coarsened grids is demonstrated in a series of numerical examples in [9]. Instead of including further numerical results here, we only state the main conclusions from [9]:

- Nonuniformly coarsened grids give significantly more accurate water-cut curves than one obtains using uniformly coarsened grids with a similar number of blocks.
- It is very easy to select parameters $V_{\min}$ and $G_{\max}$ to give a desired level of upscaling. Moreover, the accuracy of water-cut curves obtained on nonuniform coarse grids is nearly insensitive to the degree of upscaling.
- Although the nonuniform coarsening algorithm employs an initial velocity field, the coarse grid does not have to be regenerated if the flow field changes significantly, for example, if new wells are opened, or choke settings are altered. This is due to the fact that the nonuniform coarse grid essentially adapts to high permeable regions with good connectivity.
- The grid needs to be regenerated if the geology is altered significantly. However, the time spent on generating the coarse grid is usually small relative to the simulation time. The nonuniform coarsening algorithm therefore allows grids to be generated at run-time.

Hence, in combination with mixed MsFEM the nonuniform coarsening approach provides a foundation for a simulation technology that is capable of selecting grids at run-time and performing simulations in a matter of minutes, rather than hours or days. This type of simulation time may open up for using reservoir simulation for operational decision support. In the next section we discuss how the mixed MsFEM alone may be used to provide flow-based information that can be used in operational reservoir management workflows.



(a) Logarithm of permeability.

(b) $\log|v|$ projected onto the fine grid (13,200 cells).

(c) $\log|v|$ projected onto a grid with 240 blocks.

(d) $\log|v|$ projected onto a nonuniformly coarsened grid with 239 blocks.

(e) Water-cut curves

Fig. 5.21. Illustration of the nonuniform coarsening algorithm ability to generate grids that resolve flow patterns and produce accurate production estimates.

5.5.3 Utilization of multiscale methods for operational decision support

Reservoir simulation is used today as part of long term strategic planning (e.g., to predict production, quantify uncertainty, and evaluate the objective function in history matching). But to our knowledge, it is not common to utilize reservoir simulations or static flow-based information (for instance, a snapshot of the reservoir pressure and velocity fields) for operational decision support. A reason may be that traditional reservoir simulators are built as black-box tools targeting applications where only the phase saturations and production data are needed or used. Another reason may be that reservoir simulation is generally time-consuming and limited to low-resolution models, or localized high-resolution models. Hence, using reservoir simulation for decision support is not regarded as an option when decisions need to be made on a daily, hour-by-hour, or minute-by-minute basis.

With state-of-the-art multiscale techniques it is possible to evaluate flow responses of suggested well locations almost instantly. These techniques can of course also be used to run very fast reservoir simulations, but to release the full potential of multiscale methods one should not see them only as tools to accelerate simulations. Indeed, because multiscale methods can provide (accurate) information about flow patterns almost instantly, also on large-scale high-resolution models, they may have a huge potential for improving current decision support tools and work processes where flow information is not used, for example, due to too long response-time of conventional simulators. Using a multiscale solver will be particularly efficient if the flow field needs to be updated due to small or localized changes in the reservoir parameters, well configuration, and so on. Then, all that is needed is to update a few local basis functions to reflect changes in reservoir properties and so on, before the global flow can be solved very efficiently on a relatively coarse grid in (less than) a few seconds.

In the following we discuss various ways of using a snapshot of the reservoir velocity field to extract information that we believe can be valuable in operational reservoir management. Examples of information that can be extracted from a snapshot of the velocity field include:

- Injector–producer dependence
- Estimated well-sweep, that is, regions flooded by each injector
- How the flow changes by altering choke settings or inserting new wells
- How the flow is affected by perturbing the geology

One option for utilizing a velocity solution for these applications is to map streamlines (lines tangential to the velocity). This option is available today with commercial streamline simulators (e.g., FrontSim and 3DSL) or certain geomodeling tools (e.g., IRAP). Although streamline tracing scales very well with model sizes, current methods for solving the flow field do not. Utilizing a multiscale pressure solver will improve the scaling dramatically and open up significantly faster response times or larger model sizes.

For complex reservoirs with strong heterogeneity, many wells, and/or a large number of faults, streamlines typically form complex intertwined bundles and it can be difficult for the naked eye to distinguish the different well-sweep regions. In addition, accurate tracing of streamlines on models with complex grid geometry is nontrivial. For such cases, it may be more advantageous to provide information about reservoir partitioning and communication patterns in terms of volumetric objects that are bounded by surface patches or consist of a collection of grid cells. In the following, we present means to provide much of the same information one can extract from a streamline map directly on the physical grid in a way that is easy to compute and visualize.

Consider two equations of the same form: the time-of-flight equation

$$v \cdot \nabla \tau = \phi_{\mathrm{por}}, \quad \tau(\cup w_i^{\mathrm{in}}) = 0, \tag{5.36}$$

where w_i^{in} denotes injection well i, and the stationary tracer equation

$$v \cdot \nabla c_i = 0, \quad c_i(w_i^{\mathrm{in}}) = 1. \tag{5.37}$$

Here c_i models the eventual concentration of a tracer if released continually from injection well i, that is, if the injected substance is a unique tracer.

Presuming now that the velocity v is known, the time-of-flight equation (5.36) and the tracer equations (5.37) can be solved efficiently using an upstream-weighted discontinuous Galerkin (dG) method [206]. For instance, to compute the time-of-flight τ using a first-order upstream weighted dG method, we solve the following system of equations,

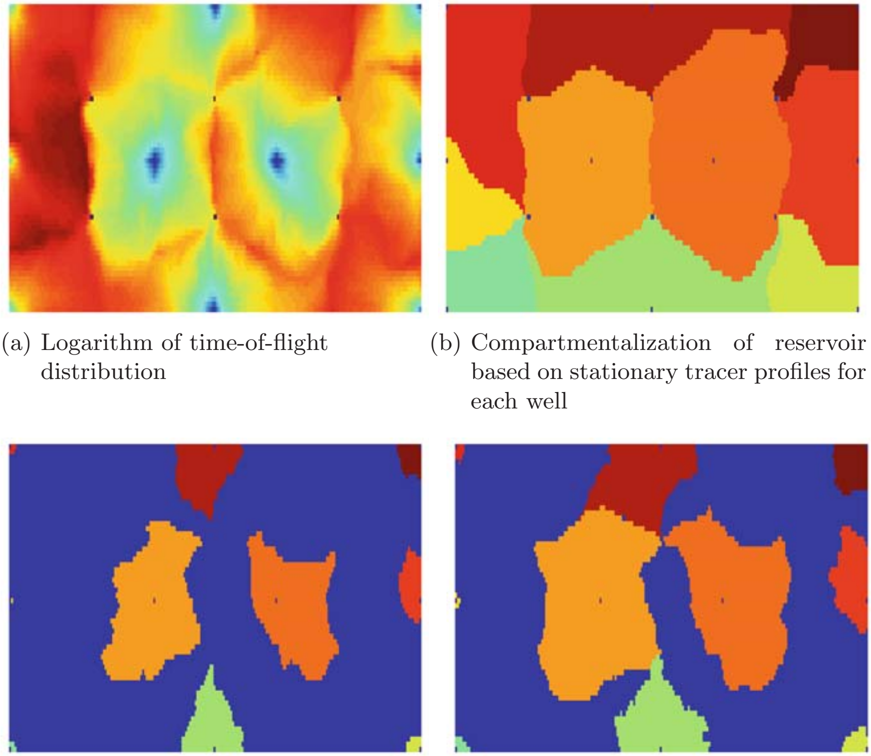
$$\int_{\partial T_i} \tau^+ v \cdot n \, ds = \int_{T_i} \phi_{\mathrm{por}} \, dx, \tag{5.38}$$

for all cells T_i. Here τ is a cellwise constant function, n is the unit normal on ∂T_i pointing outward, and τ^+ is τ evaluated on the upstream side of each interface; that is

$$\int_{\partial T_i} \tau^+ v \cdot n \, ds = \sum_j \left(\tau(T_i) \max\{v_{ij}, 0\} + \tau(T_j) \min\{v_{ij}, 0\} \right),$$

where v_{ij} is the flux from T_i to T_j. Using an optimal reordering of the cells, the discretized system can be cast as a block-triangular system that can be solved hyperfast [206].

Figure 5.22 shows time-of-flight and stationary tracer distribution for a case with ten pressure-constrained injection wells - eight along the perimeter and two in the middle — and six rate-constrained production wells. In the tracer profile plots the color of a cell corresponds to the tracer with the highest concentration. We see here that the reservoir is neatly divided into separate regions. Combining the time-of-flight information with the tracer data (i.e.,



(a) Logarithm of time-of-flight distribution

(b) Compartmentalization of reservoir based on stationary tracer profiles for each well

(c) Tracer profile with $\tau \leq 0.3$ PVI

(d) Tracer profile with $\tau \leq 0.7$ PVI

Fig. 5.22. Example of how time-of-flight and stationary tracer profiles can be used to give a visual picture of flooded regions.

by only coloring cells with $\tau \leq T$), we can easily estimate and visualize the regions that are expected to be flooded at time $t = T$ by a unit displacement front arising from each injector at time $t = 0$.

One area where plots of synthetic tracer profiles can be valuable is planning of new wells, where tracer/time-of-flight data may be used to visually inspect how adding a new or moving an existing well affects the injector-producer coupling, breakthrough times, and flooded/drained regions. It is also easy to add a mathematical measure indicating the quality of a well location. This may be useful for reservoirs with many wells where it can be difficult to assess the quality of a potential well location visually. When a new well is added or a well is moved, the flow field needs to be updated before the tracer distribution can be computed. Using mixed MsFEM with precomputed basis functions, this can be performed very efficiently by only updating the basis functions affected by the change in the given well (one basis function for each block the associated well-bore penetrates) and then assembling and solving the coarse-grid system.

With an implementation in a compiled language one should be able to recompute and visualize tracer profiles in a matter of seconds or minutes, also for large models with a large number of wells. With this type of computation time one may foresee visual and interactive ways for the early-stage optimization of the well placement, for example, by using simple trial and error. Optimization methods can also be used to semi-automate the selection of well location candidates. More accurate optimization will of course require more fine-tuned simulations. Similarly, the visual power of tracer/time-of-flight type data can be utilized in other workflows, such as ranking of multiple realizations, placements of faults, to reveal regions of interest in an assisted history-matching approach, and so on.

5.5.4 Summary

We have discussed the application of a multiscale finite element-based simulation technology for three-phase black oil reservoir simulation. In particular, we have discussed how the multiscale mixed finite element method allows faster discretization of the pressure equation in reservoir simulation, or simulation directly on high resolution geomodels.

Computation time/model size when saturation equations are solved with

- Conventional finite difference method: 10–30 times faster/larger
- Streamline methods/cell reordering: 20–100 times faster/larger
- Finite difference method on nonuniformly coarsened grid: 100– times faster/larger

We have also discussed how using MsFEMs may open up for using flow-based information for operational decision support, for instance by allowing almost instant computation and visualization of well-sweep and injector-producer pairs. Providing tools for rapid computation of this type of flow-based information can be instrumental in increasing the interactivity and reduce the turnaround time for various reservoir management workflows. In particular, to bridge the gap between the geomodel and the simulation model, it may be necessary that simulation grids and suitable simulation technology can be selected in a semi-automated manner at run-time to fit response-time requirements and available resources.

5.6 Multiscale finite volume method for black oil systems (by S. H. Lee, C. Wolfsteiner and H. A. Tchelepi)

Most practical reservoir simulation studies are performed using the so-called black oil model, in which the phase behavior is represented using solubilities and formation volume factors. We extend the multiscale finite volume (MsFV) method to deal with nonlinear immiscible three-phase compressible flow in the presence of gravity and capillary forces (i.e., black oil model). Consistent with the MsFV framework, flow and transport are treated separately and differently using a sequential implicit algorithm. A multiscale operator splitting strategy is used to solve the overall mass balance (i.e., the pressure equation). The black oil pressure equation, which is nonlinear and parabolic, is decomposed into three parts. The first is a homogeneous elliptic equation, for which the original MsFV method is used to compute the dual basis functions and the coarse-scale transmissibilities. The second equation accounts for gravity and capillary effects; the third equation accounts for mass accumulation and sources/sinks (wells). With the basis functions of the elliptic part, the coarse-scale operator can be assembled. The gravity/capillary pressure part is made up of an elliptic part and a correction term, which is computed using solutions of gravity-driven local problems. A particular solution represents accumulation and wells. The reconstructed fine-scale pressure is used to compute the fine-scale phase fluxes, which are then used to solve the nonlinear saturation equations. For this purpose, a Schwarz iterative scheme is used on the primal coarse grid. The framework is demonstrated using challenging black oil examples of nonlinear compressible multiphase flow in strongly heterogeneous formations.

5.6.1 Governing equations and discretized formulation

The standard black oil model has two hydrocarbon phases (i.e., oil and gas) and one aqueous phase (water) with rock and fluid compressibility, gravity effects, and capillary pressure. The thermodynamic equilibrium between the hydrocarbon phases is modeled via the solubility of the gas pseudo-component in the oil phase. The conservation equations are nonlinear due to the strong nonlinear character of the relative permeability and capillary pressure relations, the large gas compressibility, phase appearance and disappearance effects, and large density and viscosity differences.

The governing equations for the black oil formulation [29] are:

$$\frac{\partial}{\partial t}\left(\phi_{\text{por}} b_o S_o\right) = \operatorname{div}\left(b_o \lambda_o \left(\nabla p_o - g\rho_o e_3\right)\right) - q_o, \tag{5.39}$$

$$\frac{\partial}{\partial t}\left(\phi_{\text{por}} b_w S_w\right) = \operatorname{div}\left(b_w \lambda_w \left(\nabla p_w - g\rho_w e_3\right)\right) - q_w, \tag{5.40}$$

$$\begin{aligned} \frac{\partial}{\partial t}\left(\phi_{\text{por}}\left(b_g S_g + R_s b_o S_o\right)\right) &= \operatorname{div}\left(b_g \lambda_g \left(\nabla p_g - g\rho_g e_3\right)\right) - q_g \\ &+ \quad \operatorname{div}\left(R_s b_o \lambda_o \left(\nabla p_o - g\rho_o e_3\right)\right) - R_s q_o, \end{aligned} \tag{5.41}$$

on the domain Ω, with boundary conditions on $\partial\Omega$. Here, $\lambda_l = k(x) k_{r_l}/\mu_l$ is the mobility of phase l, where $l = o, w, g$ (i.e., oil, water, and gas); $b_l = 1/B_l$

where B_l is the formation volume factor (i.e., ratio of volume at reservoir conditions to volume at standard conditions). S_l, k_{r_l}, μ_l, ρ_l denote, respectively, the saturation, relative permeability, viscosity, and density of phase l. The well volumetric flow rate is q_l. The tensor k describes the permeability field, which is usually represented as a complex multiscale function of space. Porosity is denoted by ϕ_{por}, p_l is the phase pressure, g is gravitational acceleration, e_3 denotes the unit vector along the reservoir depth, and R_s is the solubility of gas in oil. In general, μ_l, ρ_l, B_l, R_s, and ϕ_{por} are functions of pressure. The relative permeabilities, k_{r_l}, are functions of saturation.

Saturations are constrained by $1 = S_o + S_w + S_g$, and the three phase pressures p_w, p_o, and p_g are related by two independent capillary pressure functions:

$$p_w - p_o = p_{cwo}(S_o, S_g, S_w), \quad p_g - p_o = p_{cgo}(S_o, S_g, S_w).$$

We choose the oil phase pressure as the primary variable, $p = p_o$. Multiplication of the semi-discretized equations of (5.39) to (5.41) with

$$\alpha_o = \frac{1}{b_o^{n+1}} - \frac{R_s^{n+1}}{b_g^{n+1}}, \quad \alpha_w = \frac{1}{b_w^{n+1}}, \text{ and } \alpha_g = \frac{1}{b_g^{n+1}},$$

respectively, and summation of the resulting equations gives the pressure equation:

$$L_{BO}\; p^{\nu+1} = -\frac{C_w}{\Delta t}(p^{\nu+1} - p^{\nu}) + RHS1 + RHS2,$$

where the operator for black oil is defined by

$$L_{BO} \equiv -\sum_{\ell} \alpha_\ell \operatorname{div}\left(\lambda'^{\nu}_{\ell} \nabla\right)$$

and the right-hand sides are given by

$$RHS1 = -\sum_{\ell} \alpha_\ell \operatorname{div}\left(g\rho_\ell \lambda'^{\nu}_{\ell} \nabla z\right) + \sum_{\ell=w,g} \alpha_\ell \operatorname{div}\left(\lambda'_\ell \cdot \nabla p_{c\ell o}\right)^{\nu}$$

$$RHS2 = \frac{\phi^n_{\text{por}}}{\Delta t}\left(\sum_{\ell} \alpha_\ell b^n_\ell S^n_\ell + \alpha_g R^n_s b^n_o S^n_o\right) - \frac{\phi^{\nu}_{\text{por}}}{\Delta t} - \sum_{\ell} \alpha_\ell q^{\nu}_\ell - \alpha_g (R_s q_o)^{\nu}.$$

The C_w is a weak function of pressure defined in Lee et al. [175].

5.6.2 Multiscale finite volume formulation

In the multiscale finite volume (MsFV) algorithm introduced in [159, 160, 161], the global (fine-scale) problem is partitioned into primal and dual coarse volumes as illustrated in Section 2.5.1. A set of basis functions is computed for each dual volume, and the coarse-scale problem is assembled. Using the coarse-scale system, the coarse-scale pressure is computed. The same basis functions allow for local reconstruction of the fine-scale pressure from the coarse solution.

The original MsFV algorithm [159] was designed to solve the (elliptic) pressure equation of incompressible flow in highly heterogeneous formations. The black oil model, which accounts for compressibility and capillarity, yields a nonlinear parabolic pressure equation. However, these effects are, in general, local in nature, and the pressure equation usually exhibits near-elliptic behavior. We construct a multiscale algorithm that takes advantage of this characteristic.

A multiscale, operator splitting approach is used to solve the nonlinear parabolic overall mass balance equation for the pressure field. Specifically, the black oil pressure equation is decomposed into three equations, one homogeneous and two inhomogeneous. The homogeneous (elliptic) equation is used to compute the dual basis functions and the coarse-scale transmissibilities. The first inhomogeneous part, p_g, accounts for gravity and capillarity. The second inhomogeneous equation is solved for the particular solution p_p, which accounts for accumulation (i.e., rock and fluid compressibility) and sink/source terms. Specifically, the black oil pressure equation is decomposed as follows.

$$L_{BO}\ p_h^{\nu+1} = 0, \tag{5.42}$$

$$L_{BO}\ p_g^{\nu+1} = RHS1, \tag{5.43}$$

$$L_{BO}\ p_p^{\nu+1} = -\frac{C_w}{\Delta t}[(p_h + p_g + p_p)^{\nu+1} - (p_h + p_g + p_p)^{\nu}] + RHS2. \tag{5.44}$$

Homogeneous pressure solution

The original MsFV method [159] employs locally computed basis functions (on the fine scale) and a pressure operator on a coarse grid. The fine-scale pressure field can then be obtained via a reconstruction step. Recently, Lunati and Jenny [186] presented a MsFV method for compressible multiphase flow. Their third proposed algorithm is somewhat similar to the scheme presented in this section; however, we do not use explicitly computed coarse-scale formation volume factors.

A conforming coarse grid with N nodes and M cells is constructed on the original fine grid. Each coarse cell K_c^i with $i \in \{1, ..., M\}$ is composed of multiple fine cells. A dual coarse grid is constructed such that each dual coarse cell K_d^j, $j \in \{1, ..., N\}$ contains exactly one coarse node. The coarse dual grid has M nodes, x_i ($i \in \{1, ..., M\}$), each in the interior of a coarse cell K_d^i. Each dual grid has N_c corners (for a Cartesian grid, four in two dimensions and eight in three dimensions). A set of dual basis functions, ϕ_j^i, is constructed, one for each corner i of each dual coarse cell K_d^j.

The dual basis functions are used to assemble the coarse-scale transmissibility field for computation of the coarse-scale pressure p_i^c. The dual basis function ϕ_j^i, for example, is the local solution of (5.42):

$$\alpha_o \operatorname{div}\left(\lambda'^{\nu}_o \nabla \phi^i_j\right) + \alpha_w \operatorname{div}\left(\lambda'^{\nu}_w \nabla \phi^i_j\right) + \alpha_g \operatorname{div}\left(\lambda'^{\nu}_g \nabla \phi^i_j\right) = 0 \text{ on } K^j_d,$$

where properties from the underlying fine grid (e.g., total mobility) are used. The boundary conditions are obtained by solving the reduced problems [175], although one can easily use different boundary conditions as discussed earlier. Finally, given a coarse-scale solution p^c_i, the phase transmissibilities of the coarse grid can be readily computed from the fluxes across the coarse grid interface [175].

Inhomogeneous solution: Gravity and capillary pressure

As shown in (5.43), the inhomogeneous solution p_g accounts for gravity and capillary forces. Due to the complexity of the fractional flow function in the presence of gravity, the potential field cannot be represented by a simple superposition of the basis functions. Lunati and Jenny [187] proposed a method where p_g is split into two parts. The first part is represented by the original dual basis functions; the second part is a locally computed correction term that accounts for buoyancy effects. Following their approach, p_g can be written as

$$p_g = p^a_g + p^b_g = \sum_i \phi^i_j p^{c,i}_g + p^b_g \text{ in } K^j_d. \tag{5.45}$$

Note that within a dual coarse grid, p^a_g is represented by a linear combination of basis functions, weighted by the coarse-scale pressures.

The additional correction term p^b_g is obtained using (5.43),

$$\begin{aligned}-\alpha_o \operatorname{div}\left(\lambda'^{\nu}_o \nabla p^b_g\right) - \alpha_w \operatorname{div}\left(\lambda'^{\nu}_w \nabla p^b_g\right) - \alpha_g \operatorname{div}\left(\lambda'^{\nu}_g \nabla p^b_g\right)& \\ = RHS1 \text{ in } K_d,&\end{aligned} \tag{5.46}$$

where solutions of reduced problems consistent with (5.46) serve as boundary conditions. Note that the correction term p^b_g is computed with the simple boundary conditions for the reduced system that is independent of the global pressure distribution. This particular localization assumption to compute p^b_g is analogous to the one used to construct the dual basis function in the absence of gravity effects. Lunati and Jenny [187] showed its effectiveness in resolving the fine-scale structures of complex heterogeneous problems, when buoyancy plays an important role.

Substitution of (5.45) in (5.43) and applying Green's theorem to the coarse operator [187], one can readily show that p^b_g acts as an additional source/sink term in the coarse-scale pressure system.

Particular solution: Mass accumulation and wells

The particular solution p_p, governed by (5.44), is used to model sources and sinks and the effects of compressibility (i.e., fluid accumulation). Accurate modeling of wells is crucial for any practical reservoir simulation problem. A treatment of wells specifically designed for the MsFV method has recently been proposed by Wolfsteiner, Lee and Tchelepi [259]. The framework allows for modeling wells that penetrate one or multiple fine cells, and accommodates fixed-rate or fixed-pressure operating conditions. In their approach, the near-singular pressure distribution around the well is removed by a change of variables. The well effects are approximated using special basis functions that are then added to a smoothly varying background solution computed using the standard MsFV method. Here, we employ a very simple model, where wells are represented only on the coarse grid. The corresponding fine cells receive source terms of equal strength [159].

Once the coarse-scale pressure is computed, the fine-grid pressure in the dual grid can be obtained using the basis functions.:

$$p_p(x) + p_g(x) = \sum_{i=1}^{N_c} \phi_j^i(x)(p_{p,i}^c + p_{g,i}^c) + p_g^b(x), \text{ for } x \in K_d^j. \tag{5.47}$$

The pressure from the particular solution and the linear gravity part are interpolated using the dual basis functions, and then the gravity correction term, p_g^b, is added.

Jenny, Lee and Tchelepi [159, 160] found that the fine-scale velocity field computed directly from the reconstructed pressure (i.e., using the coarse-grid solution and the dual basis functions) suffered from local mass balance errors along the dual coarse cell boundaries. As a remedy, they proposed a second set of (primal) basis functions that guarantee a conservative fine-scale velocity field. In doing so, it is critical to honor the fine-scale fluxes from the overlapping dual basis functions as boundary conditions. That approach can be expensive, however. This is because the number of primal basis functions is large, and even if they need to be recomputed occasionally, the cost can be significant.

Here, we do not use this second set of bases. Instead we solve local problems on the primal coarse grid as follows. The reconstructed fine-scale pressure at the boundaries of a primal coarse cell is used to compute the fine-scale fluxes, which then serve as boundary conditions for local problems on the primal coarse grid. These local problems solve the nonlinear black oil equations, which may include compressibility, capillarity, and solubility effects. In essence the multiscale pressure approximation is used to prescribe flux boundary conditions for the full black oil equation set on the local primal coarse-cell level. Our experience is that the fine-scale pressure solution obtained from these local Neumann problems is quite accurate (i.e., locally consistent with the velocity field) when compared to the pressure that is reconstructed using the dual basis.

5.6.3 Sequential fully implicit coupling and adaptive computation

In the algorithm presented in the previous sections, flow and transport are solved sequentially. First, a fine-scale pressure field together with a compatible (and conservative) fine-scale velocity field is computed using the black oil MsFV method. Then, the transport problem is solved on local fine-grid domains with an implicit upwind scheme. A Schwarz overlap method is used with saturation at the boundaries from the previous iteration, which has been found to be very efficient for the saturation equations. The updated saturation distribution defines a new total mobility field for the subsequent elliptic problem (i.e., the next Newton iteration). Note that, in general, some of the basis functions have to be recomputed. These steps can be iterated until convergence of all variables at the current time level.

The MsFV approach can be easily adapted to a sequential fully implicit treatment [161]. The MsFV implementation allows for performing an IMPES, traditional sequential [29], or a fully implicit scheme. Here, the full nonlinear transmissibility terms at the new time-step level are retained so that stability is guaranteed [160]. The converged solution using this sequential approach should be identical to the solution obtained using the simultaneous solution strategy, which is usually used to deal with coupled fully implicit systems.

The MsFV approach is well suited for adaptive computation, which can lead to significant efficiency gains. The most expensive part of the algorithm is computation of the dual basis functions. In general, this is performed every iteration due to changes in the saturation (mobility) field. As discussed in Jenny et al. [160], an adaptive scheme can be used to update the dual basis functions. Because the basis functions are constructed with local support, the change of the total mobility is used to decide when and where to update the basis functions in the domain. For compressible fluid, we employ an effective total mobility change criterion for adaptable computation of the pressure field [175].

5.6.4 Numerical examples

Waterflood in linear geometries

This test case is a two-dimensional problem with 220×60 fine cells. A uniform coarse grid of 22×6 is used for the multiscale run. The permeability description is taken from the first layer of the Tenth SPE Comparative Solution Project [78]. As shown in Figure 5.23(a), a highly correlated area of low permeability is found on the left-hand-side of the model, and a high-permeability area is present on the right end of the model.

The black oil model includes three compressible fluid phases (i.e., oil, water, and gas). The pressure dependence of the densities is described using formation volume factors, and the phase equilibrium between the oil and gas phases is described using the solution gas–oil ratio [79]. Typical black oil properties are listed in Lee, Wolfsteiner, and Tchelepi [175]. For this example, we did not consider gravity effects, and we used a high oil compressibility, namely, $4.8 \cdot 10^{-4}$ for $p < p_b$ (p_b is the bubble point pressure) for a stringent test of compressibility. As the pressure decreases, some solution gas is liberated from the oil phase and forms a free immiscible gas phase. Moreover, the oil-phase volume decreases as the pressure decreases below the bubble point, $p < p_b$. The solution-gas is constant above the bubble point pressure, and the oil-phase volume decreases as the pressure increases (i.e., $p > p_b$).

The model is initialized with oil ($S_o = 1$) and constant pressure (4000 psia). At $t = 0$, water is injected at a constant pressure of 5000 psia from the left side; the right boundary is maintained at 2000 psia. This numerical example is a challenging test due to the large pressure drop across the model and the large variations in permeability. A constant time-step size of 1 day is used. In Figure 5.23, the results from the black oil MsFV simulator and fine-scale reference simulations are shown at 50 days. For example, Figure 5.23(c)-(d) indicate that the water and gas distributions obtained from the MsFV approach are in excellent agreement with the reference fine-scale solutions.

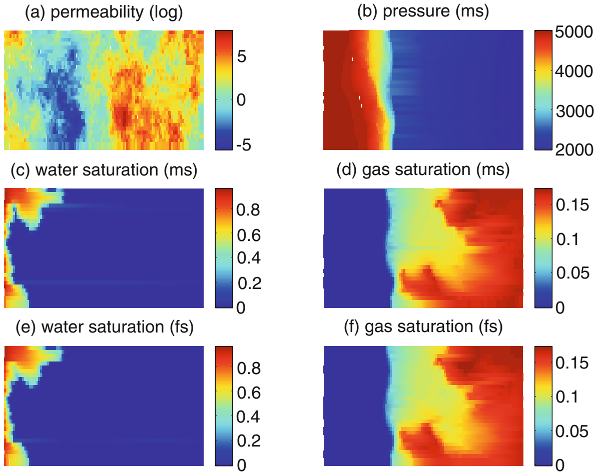


Fig. 5.23. Depletion with constant pressure boundary conditions: at time = 50days: (a) logarithm of permeability; (b) pressure from MsFV; (c) S_w from MsFV,(d) S_g from MsFV; (e) S_w from fine-scale simulation; (f) S_g from fine-scale simulation.

A three-dimensional heterogeneous model with two wells

This example employs a three-dimensional model with two wells at two opposite corners and a heterogeneous permeability field. The permeability distribution is generated by the sequential Gaussian simulation method [85]. The logarithm of permeability has a Gaussian histogram with mean and standard deviation of 50 md and 1.5, respectively. The variogram is spherical with ranges of 30 m and 15 m in directions that are at 45 and 135 degrees with respect to the horizontal, and 7.5 m in the vertical direction. The permeability is shown in Figure 5.24(a). The model is 150 × 150 × 48 m in size and is uniformly discretized using $45 \times 45 \times 30$ fine cells. The uniform $9 \times 9 \times 6$ coarse grid is used in the MsFV computations.

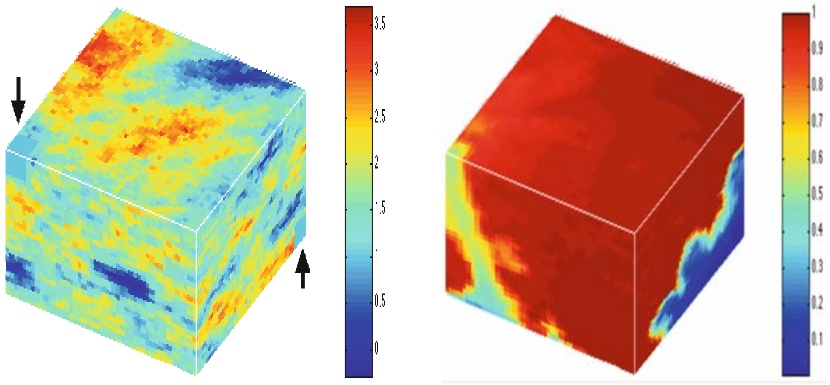
The permeability distribution is shown in Figure 5.24(a). The fluid properties for the first example are also employed. The reservoir is initially at gravitational equilibrium with 4000 psia at the bottom of the model. Water is injected at a constant rate from the bottom left corner (i.e., coarse cell 1,1,1) displacing the oil toward the producer located at the top right corner (cell 9,9,6). Figure 5.24(b) shows the oil saturation distribution at water breakthrough. The pressure around the production well is below the bubble point, and a free gas phase is present. In Figure 5.25, the production rates from MsFV are compared with those from the fine-scale reference simulation. The comparison shows that the black oil MsFV approach is able to model difficult multiphase flow problems in heterogeneous media when strong gravity and compressibility effects are present.

We also performed computations using an upscaled model for this problem. We used the basis functions to compute an upscaled (effective coarse-scale) transmissibility field, and we computed the pressure and saturation using the coarse-scale model. The results are also depicted in Figure 5.25. Even though the results from the upscaled model are qualitatively similar to those from the fine-scale reference simulation, the presence of large numerical dispersion in the upscaled model gives less accurate production rates compared with the multiscale method. This numerical example shows that reconstruction of the fine-scale information by the MsFV is an important step in obtaining accurate transport predictions.

5.6.5 Remarks

We developed a multiscale finite volume (MsFV) method for the black oil formulation of multiphase flow and transport in heterogeneous porous media. The black oil formulation, which involves immiscible three-phase flow with compressibility, gravity, capillary, and mass transfer, in the form of gas solubility, is widely used in practical field-scale simulations.

Our approach extends the sequential implicit MsFV method [161, 256] to the nonlinear black oil model. An operator-splitting multiscale algorithm is devised to compute the fine-scale pressure field, which is used to compute the fine-scale velocity field. The nonlinear saturation equations of the black oil model are solved on the local primal coarse grid using the fine-scale velocity field. The black oil MsFV method extends our ability to deal with large-scale problems of practical interest. The treatment ensures that the nonlinearity due to rock and fluid compressibility, gravity, and capillarity can be resolved



(a) (b)

Fig. 5.24. A heterogeneous model with two wells (Example 3): (a) log-permeability distribution, (b) oil saturation just after breakthrough.

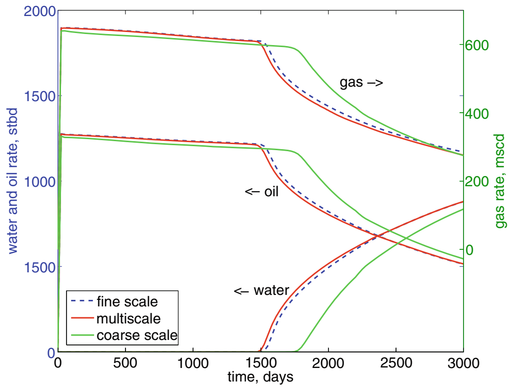


Fig. 5.25. Comparison of production rates for Example 3: multiscale (solid lines) fine-scale reference (dashed).

by solving specially constructed local boundary value problems. The methodology is demonstrated using several numerical examples. These examples show clearly that the MsFV scheme yields results that are in excellent agreement with reference fine-grid solutions.

Although the numerical efficiency of this new black oil MsFV simulator has not been fully examined, the numerical efficiency gains shown in references [160, 161, 256] are expected to hold (e.g., 10 $\sim$ 20 times faster than the conventional finite difference method). This is because all the nonlinearities due to the presence of compressibility, gravity, and capillary pressure are resolved locally.

5.7 Applications of multiscale finite element methods to stochastic flows in heterogeneous media

The media properties often contain uncertainties. These uncertainties are usually parameterized and one has to deal with a large set of permeability fields (realizations). This brings an additional challenge to the fine-scale simulations and necessitates the use of coarse-scale models. The multiscale methods are important for such problems. In this section, we describe the extensions of MsFEMs to stochastic equations where the basis functions are constructed such that they span both spaces and uncertainties. We also consider the applications of MsFEMs to uncertainty quantification in inverse problems when the media properties are estimated based on coarse-scale data.

First, we briefly discuss stochastic flow equations from an application point of view. Assume that the media properties are random and denoted by $k(x,\omega)$, where ω refers to a realization. Then, the solution of the flow equation is given by $p(x,\omega)$ for each realization ω.

One of the commonly used stochastic descriptions of spatial fields is based on a two-point correlation function of log-permeability. To describe it, we denote by $Y(x,\omega)=\log[k(x,\omega)]$. For permeability fields described with the two-point correlation function, it is assumed that $R(x,y)=E\left[Y(x,\omega)Y(y,\omega)\right]$ is known, where $E[\cdot]$ refers to the expectation (i.e., average over all realizations) and x,y are points in the spatial domain. In applications, the permeability fields are considered to be defined on a discrete grid. In this case, $R(x,y)$ is a square matrix with N_{dof} rows and N_{dof} columns, where N_{dof} is the number of grid blocks in the domain. For permeability fields described by the two-point correlation function, one can use the Karhunen–Loève expansion (KLE) [182, 271] to obtain a permeability field description with possibly fewer degrees of freedom. This is done by representing the permeability field in terms of an optimal L^2 basis. By truncating the expansion, we can represent the permeability matrix by a small number of random parameters.

We briefly recall some properties of the KLE. For simplicity, we assume that $E[Y(x,\omega)]=0$. Suppose $Y(x,\omega)$ is a second-order stochastic process with $E\int_\Omega Y^2(x,\omega)dx<\infty$. Given an orthonormal basis $\{\Phi_i\}$ in $L^2(\Omega)$, we can expand $Y(x,\omega)$ as a general Fourier series

$$Y(x,\omega)=\sum_{i=1}^{\infty}Y_i(\omega)\Phi_i(x), \qquad Y_i(\omega)=\int_\Omega Y(x,\omega)\Phi_i(x)dx.$$

We are interested in the special L^2 basis $\{\Phi_i\}$ that makes the random variables Y_i uncorrelated. That is, $E(Y_iY_j)=0$ for all $i\neq j$. The basis functions $\{\Phi_i\}$ satisfy

$$E[Y_iY_j]=\int_\Omega \Phi_i(x)dx\int_\Omega R(x,y)\Phi_j(y)dy=0, \quad i\neq j.$$

Because $\{\Phi_i\}$ is a complete basis in $L^2(\Omega)$, it follows that $\Phi_i(x)$ are eigenfunctions of $R(x,y)$:

$$\int_{\Omega} R(x,y)\Phi_i(y)dy = \lambda_i \Phi_i(x), \quad i = 1, 2, \ldots, \tag{5.48}$$

where $\lambda_i = E[Y_i^2] > 0$. Furthermore, we have

$$R(x,y) = \sum_{i=1}^{\infty} \lambda_i \Phi_i(x)\Phi_i(y). \tag{5.49}$$

Denote $\theta_i = Y_i/\sqrt{\lambda_i}$; then θ_i satisfy $E(\theta_i) = 0$ and $E(\theta_i\theta_j) = \delta_{ij}$. It follows that

$$Y(x,\omega) = \sum_{i=1}^{\infty} \sqrt{\lambda_i}\theta_i(\omega)\Phi_i(x), \tag{5.50}$$

where Φ_i and λ_i satisfy (5.48). We assume that the eigenvalues λ_i are ordered as $\lambda_1 \geq \lambda_2 \geq \cdots$. The expansion (5.50) is called the Karhunen–Loève expansion. In the KLE (5.50), the L^2 basis functions $\Phi_i(x)$ are deterministic and resolve the spatial dependence of the permeability field. The randomness is represented by the scalar random variables θ_i. After we discretize the domain Ω by a rectangular mesh, the continuous KLE (5.50) is reduced to finite terms and $\Phi_i(x)$ are discrete fields. Generally, we only need to keep the leading order terms (quantified by the magnitude of λ_i) and still capture most of the energy of the stochastic process $Y(x,\omega)$. For an N-term KLE approximation $Y_N = \sum_{i=1}^{N} \sqrt{\lambda_i}\theta_i\Phi_i$, define the energy ratio of the approximation as

$$e(N) := \frac{E\|Y_N\|^2}{E\|Y\|^2} = \frac{\sum_{i=1}^{N} \lambda_i}{\sum_{i=1}^{\infty} \lambda_i}.$$

If $\lambda_i, i = 1, 2, \ldots$, decay very fast, then the truncated KLE would be a good approximation of the stochastic process in the L^2 sense.

Next, we discuss some example cases. Suppose the permeability field $k(x,\omega)$ is a log-normal homogeneous stochastic process; then $Y(x,\omega) = \log(k(x,\omega))$ is a Gaussian process, and θ_i are independent standard Gaussian random variables. In this case, the covariance function of $Y(x,\omega)$ has the form

$$R(x,y) = \sigma^2 \exp\Big(-\frac{|x_1 - y_1|^2}{2l_1^2} - \frac{|x_2 - y_2|^2}{2l_2^2}\Big). \tag{5.51}$$

In the above formula, l_1 and l_2 are the correlation lengths in each dimension, and $\sigma^2 = E(Y^2)$ is the variance. We first solve the eigenvalue problem (5.48) numerically on the rectangular mesh and obtain the eigenpairs $\{\lambda_i, \Phi_i\}$. We put 8 points per correlation length in our numerical simulations. Because the eigenvalues decay fast, the truncated KLE approximates the stochastic process $Y(x,\omega)$ fairly well in the L^2 sense. Therefore, we can sample $Y(x,\omega)$ from the truncated KLE (5.50) by generating Gaussian random variables θ_i. In Figure 5.26, we plot eigenvalues and three eigenvectors corresponding to eigenvalues (in decreasing order) 1, 6, and 15. In particular, we plot eigenvalues for the log-normal permeability field described by (5.51) as well as by

$$R(x,y) = \sigma^2 \exp\Big(-\frac{|x_1 - y_1|}{l_1} - \frac{|x_2 - y_2|}{l_2}\Big). \tag{5.52}$$

As we see from these figures the eigenvalues decay quickly for log-normal permeability fields compared to log-permeability fields described by (5.52). Moreover, the eigenvectors corresponding to smaller (in value) eigenvectors contain finer-scale features of the media.

For some simplified cases, one can derive formulas for eigenvalues and eigenvectors (e.g., [277]). In the one-dimensional case, $R(x,y) = \sigma^2 \exp(-|x_1 - y_1|/l_1)$, the eigenvalues have the form

$$\lambda_n = \frac{2l_1\sigma^2}{l_1^2\zeta_n^2 + 1}$$

and

$$\Phi_n(x) = \frac{1}{\sqrt{(l_1^2\zeta_n^2 + 1)L/2 + l_1}}(l_1\zeta_n\cos(\zeta_n x) + \sin(\zeta_n x)),$$

where L is the length of the domain and ζ_n are positive roots of the characteristic equation

$$(\zeta^2 l_1^2 - 1)\sin(\zeta L) = 2\zeta l_1 \cos(\zeta L). \tag{5.53}$$

For problems in a multidimension, if the covariance function is in the form $R(x,y) = \sigma^2 \exp(-|x_1 - y_1|/l_1 - |x_2 - y_2|/l_2)$, the eigenvalues have the form

$$\lambda_{ij} = \frac{4l_1 l_2\sigma^2}{(l_1^2(\zeta_i^1)^2 + 1)(l_2^2(\zeta_j^2)^2 + 1)}$$

and

$$\Phi_{ij}(x) = \Phi_i(x_1)\Phi_j(x_2),$$

where ζ_i^1 and ζ_j^2 are positive roots of (5.53) using parameters (L_1, l_1) and (L_2, l_2), respectively, with L_1 and L_2 being the lengths of the whole domain in the x_1- and x_2-directions.

5.7.1 Multiscale methods for stochastic equations

In this section, we present a multiscale approach for solving stochastic flow equations. The main idea of the proposed approaches is to construct multiscale basis functions that capture the small-scale information across the realizations of stochastic equations. Once the basis functions are constructed, the solution is projected into the finite-dimensional space spanned by the multiscale basis functions. The pre-computed basis functions are constructed based on selected realizations of the stochastic permeability field and the method can be regarded as an extension of MsFEMs to stochastic porous media equations. The proposed approaches, although they do not require any interpolation in stochastic space, can be combined with interpolation-based approaches to

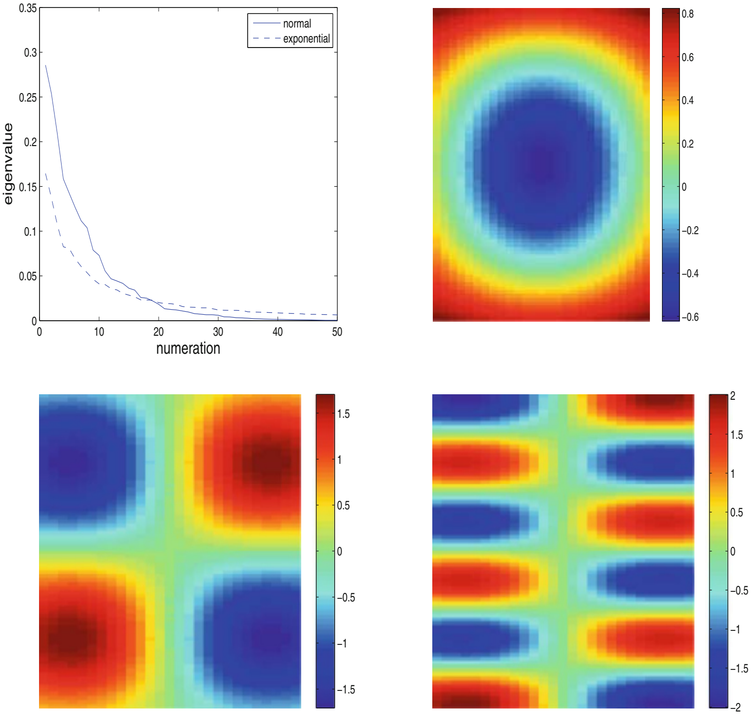


Fig. 5.26. Top left: Eigenvalue distribution. Top right: 1st eigenvector. Bottom left: 6th eigenvector. Bottom right: 15th eigenvector.

predict the solution on the coarse grid. The permeability fields under consideration do not have scale separation. For this reason, we employ multiscale methods using limited global information in our simulations. The main idea of these approaches is to use some global fields that contain nonlocal information as discussed in Chapter 4. We use the mixed MsFEM framework here, although other global couplings can also be used.

To present the approach, we consider realizations of permeability fields k_i sampled from a stochastic distribution. For each k_i, let p_i and v_i be a solution obtained by solving the flow equation on a fine grid using a suitable mass conservative numerical method. Then, we define $\mathcal{V}_h(k_i)$ the space spanned by mixed multiscale basis functions, ψ_{il}^K, defined via (4.6) that have the following boundary conditions

$$\psi_{il}^K \cdot n = \frac{v_i \cdot n}{\int_{e_l} v_i \cdot nds},$$

on e_l for each K. We employ N realizations of the permeability field and define a finite-dimensional space that consists of a direct sum of mixed multiscale finite element basis functions corresponding to all realizations:

$$\mathcal{V}_h = \oplus_{i=1}^{N} \mathcal{V}_h(k_i).$$

Hence, in this case we obtain N basis functions for each edge (face) in the coarse grid. Once the basis functions are constructed, the solution of the stochastic flow equation for an arbitrary realization is projected onto this finite-dimensional space. Note that this approach does not require any interpolation formula in uncertainty space, although interpolation techniques, if easily available, can be used to reduce the size of $\mathcal{V}_h$ locally in uncertainty space (see later the use of interpolation techniques). We assume that $v_1, ..., v_N$ are linearly independent in order to guarantee that the basis functions are linearly independent. Note that the local basis functions can be used in the proposed multiscale approach for stochastic flow equations.

Next, we present a formal analysis of the method under the assumption that the chosen realizations can be used to interpolate an arbitrary realization. To show this, we assume that the uncertainties of the permeability field can be parameterized. As a result of this parameterization, the permeability is expressed as $k = k(x, \theta)$ where $\theta \in \mathbb{R}^L$. One such example is the Karhunen–Loève expansion (KLE) as discussed earlier. KLE can be used in representing the permeability fields given via the two-point correlation function, where $k(x, \theta) = \exp(Y(x, \theta))$, $Y(x, \theta) = \sum_{i=1}^{L} \varTheta_i \varPhi_i(x)$, $\varPhi_i(x)$ pre-determined functions, and $\theta = (\varTheta_1, ..., \varTheta_L)$.

When the uncertainties are parameterized and L is not large, one can employ sparse interpolation techniques in $\mathbb{R}^L$ (e.g., [272]), where the solution is computed for some values of $\theta = (\varTheta_1, ..., \varTheta_L)$, denoted by θ_k, and then interpolated for an arbitrary $\theta \in \mathbb{R}^L$. Assuming that $k(x, \theta)$ smoothly depends on θ, we can approximate the solution for an arbitrary θ as

$$p(x, \theta) \approx \sum_i p(x, \theta_i) \beta_i(\theta), \tag{5.54}$$

where $\beta_i(\theta)$ are the corresponding weights which are in general difficult to obtain. We note that the interpolation error depends on the choice of interpolation points and the smoothness of $p(x, \theta)$ with respect to θ. Denoting the velocity field for two-phase flow by v, we have

$$v(x, \theta) \approx \sum_i v(x, \theta_i) \beta_i(\theta). \tag{5.55}$$

Equation (5.54) shows that the solution of the stochastic flow equation can be approximated if we provide approximations of $p(x, \theta_i)$ for each θ_i. Because the solution for each selected realization can be approximated using corresponding global fields, we have

$$v(x, \theta_i) \approx \sum_j c_{ij}^*(x) v_j(x, \theta_i).$$

In our numerical simulations, we use single-phase velocity fields following previous discussions in Section 4.2 (see also [1, 3]). One can, in general, use directional flows as proposed in a more general setting in [218]. We note that in our multiscale simulations, the basis functions are constructed using $v_j(x, \theta_i)$. One can show the convergence of the proposed approach following, for example, [8].

We note that the proposed method can be applied in a local region of the uncertainty space by selecting realizations that correspond to this region. The latter is useful when one would like to perform uncertainty quantification in a subregion of the uncertainty space. One can use the localization in the uncertainty space for more accurate probabilistic estimations by partitioning the uncertainty space. To our best knowledge, the idea of local partitioning of uncertainty space in the context of stochastic PDEs was first investigated in [267] where the authors introduced a multi-element generalized polynomial chaos approach. In our approaches, we can borrow this idea and combine it with MsFEMs. To describe the procedure, we denote by U the uncertainty space and assume that U is partitioned into U_i. In each region U_i, we choose selected realizations θ_j^i representing these local regions. Then, the basis functions are defined as before for these selected realizations in each U_i. This approach is an implementation of the earlier proposed technique simply in local regions of uncertainty space. In particular, the multiscale basis functions are constructed as before although with local support both in spatial and uncertainty spaces. When performing simulations for a particular (arbitrary) realization, the multiscale basis functions from the local uncertainty region that contains this particular realization will be used. This will provide high accuracy and reduce the computational cost.

We note that the proposed method can be applied in local regions of uncertainty space and, consequently, the support of basis functions can be localized in uncertainty space. To describe the procedure, we denote by U the uncertainty space and assume that U is partitioned into U_i. Here U_i can be regions larger than the characteristic length scale in uncertainty space. Furthermore, in each region U_i, we choose realizations θ_j^i representing these local regions. Then, the basis functions are defined as before for these realizations in each U_i. This approach is an implementation of the earlier proposed technique simply in local regions of uncertainty space. In particular, the multiscale basis functions are constructed as before although with local support. One can draw a parallel between this approach and a general multiscale approach where the coefficients strongly vary with respect to spatial variables and uncertainties. In particular, we would like to construct multiscale basis functions for permeability fields $k(x, \theta)$ over a coarse region that is larger than spatial and uncertainty heterogeneities. In this case, one needs to construct the local spatial basis functions for each θ_j^i in U_i. Because the dependence on θ is parametric, one needs to capture the spatial heterogeneities for all values of θ in U_i. In this case, the basis functions are derived from the solution of

$$\operatorname{div}(k(x,\theta)\nabla w_i^K(x,\theta)) = 0.$$

These basis functions, which are smooth with respect to θ, can be approximated by choosing appropriate realizations. Thus, the proposed approach can be regarded as an extension of the mixed MsFEM to problems with uncertainties. When using local patches in uncertainty space, one needs to determine a partition to which a particular realization belongs. We note that pre-computed multiscale basis functions can be repeatedly used for different boundary conditions/source terms and for dynamic two-phase flow and transport simulations.

The main practical advantage of the proposed mixed MsFEM is that one does not need interpolation formulas. Indeed, when an approximation space consists of a union of subspaces generated using the solutions corresponding to different permeability realizations, one is actually projecting the true solution onto this enriched approximation space. Thus, the velocity solution will be a superposition of basis functions corresponding to each of the sample fields, but the interpolation weights are determined automatically from the projection property of the mixed MsFEM. In particular, the interpolation weights will vary throughout the uncertainty domain. This approach is interpolation-free, easy to use, and provides a computationally cost-efficient methodology for performing multiple simulations, for instance, to quantify uncertainty. We also note that when an interpolation formula is easily available, one can interpolate the set of pre-computed multiscale basis functions to calculate the basis functions for a particular realization. However, the nature of this interpolation (pointwise or L^2 or so on) will be pre-determined. Our proposed approach chooses the best interpolation both in spatial and stochastic space. Finally, we would like to note that in upscaling approaches, to our best knowledge, one cannot avoid interpolation techniques.

Numerical results

Experimental setup. In our simulations below, we take $k_{rw}(S) = S^2$, $\mu_w = 0.1$, $k_{ro}(S) = (1-S)^2$, and $\mu_o = 1$ in two-phase flow and transport simulations (see (5.1), (5.2)). The log-permeability field $Y(x)$ is given on a 100×100 fine Cartesian grid. This grid is then coarsened to form a uniform 5×5 Cartesian grid so that each block in the coarse grid contains a 20×20 cell partition from the fine grid. We solve the pressure equation on the coarse grid using the mixed MsFEM and then reconstruct the fine-scale velocity field as a superposition of the multiscale basis functions. The reconstructed field is used to solve the saturation equation on the fine grid. The saturation equation is solved using an implicit upstream finite volume (discontinuous Galerkin) method. We would like to emphasize that the multiscale basis functions are constructed at time zero, that is, they are not recomputed during the simulations.

In the numerical examples that are reported below we consider a traditional quarter-of-a-five-spot problem. That is, Ω is taken to be a square domain, we inject water at the upper left corner, and produce the fluid that reaches the producer at the lower right corner. To assess the quality of the respective saturation solutions obtained using the mixed MsFEM, we compute for each realization a reference solution S_{ref} obtained by solving the time-dependent pressure equation on the fine grid with the given permeability field (using the lowest-order Raviart–Thomas mixed finite element method for Cartesian grids). Then, in addition to measuring the relative saturation error in the L^1-norm:

$$\|S - S_{\text{ref}}\|_{L^1} / \|S_{\text{ref}}\|_{L^1},$$

we compare various production characteristics. We use the water-cut curve defining the fraction of water in the produced fluid as a function of time measured in pore volumes injected (PVI) (see (2.44)). We recall that

$$w(t) = \frac{q_w(t)}{q_w(t) + q_o(t)},$$

where q_o and q_w are flow rates of oil and water at the producer at time t.

We monitor the following quantities

- The relative water-cut error in the L^2-norm:

$$\|w - w_{\text{ref}}\|_{L^2} / \|w_{\text{ref}}\|_{L^2}.$$

- The breakthrough time (defined as $w^{-1}(0.05)$) at the producer.
- The cumulative oil production at 0.6 PVI:

$$Q_o = -\frac{1}{\int_\Omega \phi \, dx} \int_0^{0.6PVI} \left(\int_\Omega \min(q_o(x,\tau), 0) \, dx \right) d\tau.$$

Before we embark on the numerical experiments, we note that the Raviart–Thomas mixed finite element discretization of the pressure equation results in a linear system with $N_{\text{fine}}^2 + 2 \times (N_{\text{fine}} - 1) \times N_{\text{fine}} = 29800$ unknowns, where $N_{\text{fine}} = 100$. In comparison, when using a sample of N permeability fields to generate the mixed MsFEM basis functions, the stochastic multiscale method gives rise to a linear system with $N_{\text{coarse}}^2 + 2 \times (N_{\text{coarse}} - 1) \times N_{\text{coarse}} \times N = 25 + 40N$ unknowns, where $N_{\text{coarse}} = 5$. Hence, when using a sample size of 25, for instance, the number of the unknowns in the fine-grid system is roughly 30 times larger than the number of unknowns in the mixed MsFEM system. In this section, we present our results which employ samples of 10–50 permeability fields. In other words, we compute 10–50 velocity basis functions for each interface in the coarse grid. We note that in order for the proposed methods to be computationally efficient one needs to use fewer basis functions in each coarse-grid block to represent the heterogeneities across space and uncertainties. In particular, the number of basis functions needs to be less than the number of fine-grid blocks within the target coarse-grid block. Otherwise, one can simply use fine-scale basis functions which are the same for an arbitrary realization. In the case of the latter, the stochasticity does not affect the choice of the finite element function space.

Gaussian fields. For Gaussian fields, one can reduce the dimension of the uncertainty space dramatically due to the fast decay of eigenvalues. To sample the realizations that are used to generate the multiscale basis functions, we use the first order Smolyak collocation points θ_i in $[-3, 3]^L$ (see, e.g., [272]). That is, $\theta_0 = 0$, $\theta_{2i-1} = 3\delta_{ij}$, and $\theta_{2i} = -3\delta_{ij}$, $i = 1, ..., L$. We note that the choice of interpolation points does not affect the implementation of our approach.

Our first results are for the isotropic case with $l_1 = l_2 = 0.2$ and $\sigma^2 = 2$. In this case, we can reduce the dimension of the stochastic permeability to 10. From this stochastic model for the permeability we draw randomly 100 realizations and perform simulations on the corresponding permeability fields.

In Figure 5.27 we compare breakthrough times and cumulative oil production at 0.6 PVI. We see that there is nearly a perfect match between the results obtained with the mixed MsFEM and the corresponding results derived from the reference solutions. Next, in Figure 5.28, we plot L^2 errors in the saturation field for these realizations as well as the water-cut errors. It can be observed from this figure that the saturation errors are mostly below 3%. Finally, we plot in Figure 5.29 a histogram of the breakthrough times and cumulative oil production values depicted in Figure 5.27 to demonstrate that the mixed MsFEM essentially provides the same statistics as one obtains from the set of reference solutions. These results suggest that with a few precomputed basis functions in each coarse grid block we can solve two-phase flow equations on the coarse grid for an arbitrary realization and obtain nearly the same results as one obtains by doing fine-grid simulations for each realization.

We have also considered numerical results for an anisotropic Gaussian field with $l_1 = 0.5$, $l_2 = 0.1$, and $\sigma^2 = 2$ in [7]. Due to anisotropy, KLE requires 12 terms. We sample the realizations that are used to generate the multiscale basis functions using the first order Smolyak collocation points as in the isotropic case. The numerical results obtained for the anisotropic Gaussian fields are qualitatively the same as the results shown in Figure 5.27 – Figure 5.29. We include only the anisotropic equivalent of Figure 5.29. Histograms of breakthrough time and cumulative oil production at 0.6 PVI for 100 randomly chosen realizations are depicted in Figure 5.30. The histograms confirm that the multiscale method essentially provides the same breakthrough time and cumulative oil production statistics as one obtains from the set of reference solutions.

Exponential variogram fields. For our second set of results, we consider permeability fields with exponential covariance matrix

$$R(x, y) = \sigma^2 \exp\Big(-\frac{|x_1 - y_1|}{l_1} - \frac{|x_2 - y_2|}{l_2}\Big). \tag{5.56}$$

Because of the slow decay of eigenvalues, one usually needs to keep many terms in KLE and deal with a large uncertainty space. To approximate the permeability fields, KLE requires 300 to 400 eigenvectors depending on correlation lengths and variance. This is a large-dimensional problem for performing

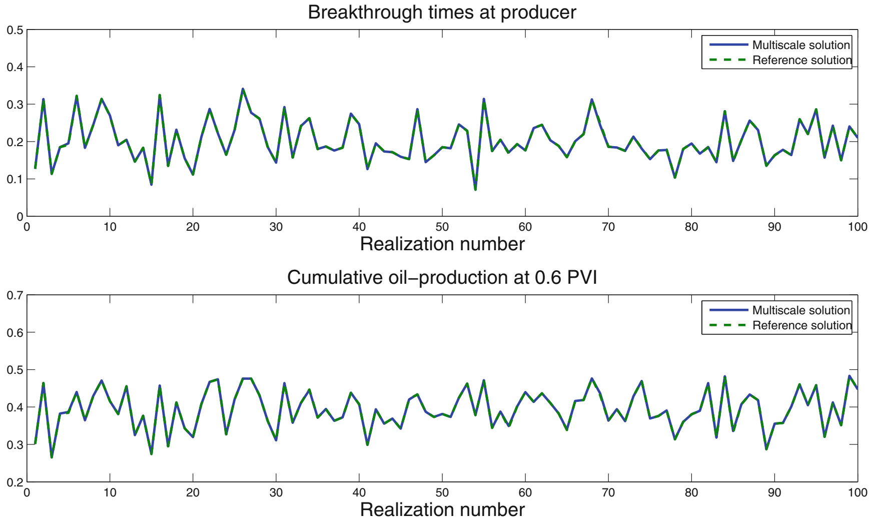


Fig. 5.27. Breakthrough time and cumulative oil production at 0.6 PVI for 100 random realizations from a Gaussian field with $l_1 = l_2 = 0.2$ and $\sigma^2 = 2$.

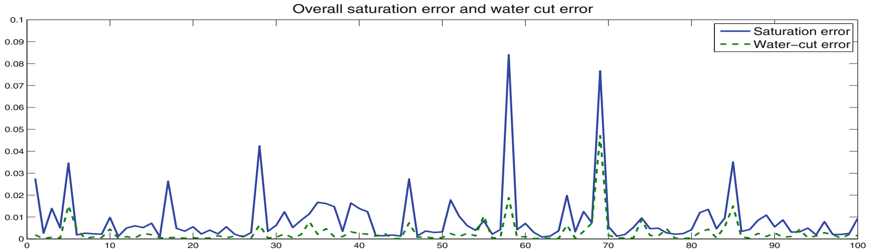


Fig. 5.28. L^2 errors of the saturation field and water-cut errors for 100 randomly chosen realizations (the number of a realization is indicated along the horizontal axis). Gaussian field with $l_1 = l_2 = 0.2$, $\sigma^2 = 2$.

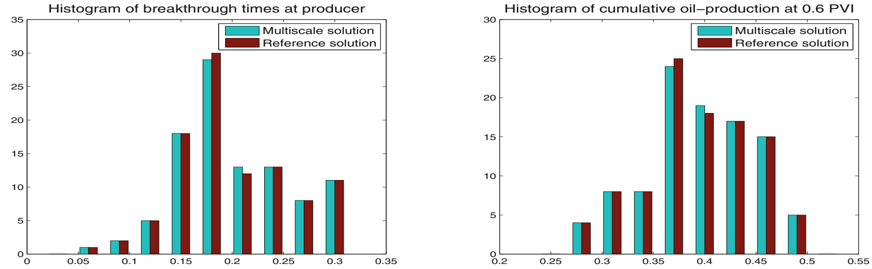


Fig. 5.29. Histograms of the breakthrough times and cumulative oil production values shown in Figure 5.27.

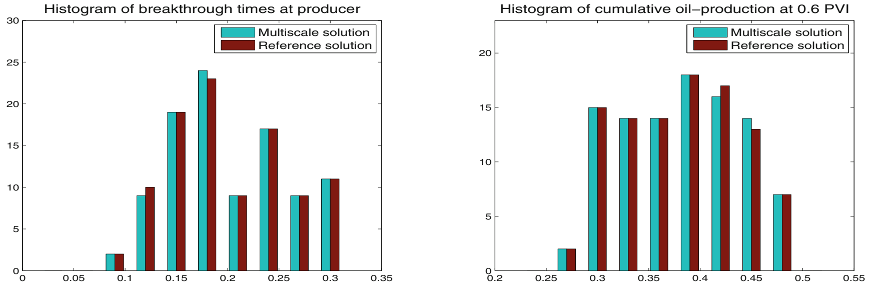


Fig. 5.30. Histograms of breakthrough time and cumulative oil production at 0.6 PVI for 100 random Gaussian fields with $l_1 = 0.5$ and $l_2 = 0.1$ and $\sigma^2 = 2$.

direct interpolation using multiscale basis functions. Instead we suggest using few independent realizations in constructing basis functions, and then performing statistical studies on a much larger set of realizations. We note that for independent realizations, we do not have an easily available interpolation formula. Moreover, the use of independent realizations is quite easy and one can use this technique for more general permeability fields in as much as it only requires independent samples of the permeability field.

To demonstrate the performance of the stochastic multiscale method for these fields, we present results for a case where the permeability fields are drawn from an anisotropic exponential variogram distribution with $l_1 = 0.5$, $l_2 = 0.1$, and $\sigma^2 = 2$ (the results for the isotropic case are similar, and not reported here). The KLE requires 350 eigenvectors to represent this stochastic permeability distribution. From this distribution we sample 20 independent realizations and use these realizations to generate the multiscale basis functions. Figure 5.31 displays one randomly chosen realization and corresponding saturation profiles at 0.6 PVI obtained by solving the pressure equation on the fine grid, and on the 5×5 coarse grid with the mixed MsFEM, respectively.

Figures 5.32, 5.33, and 5.34, show: breakthrough time at producer and cumulative oil production at 0.6 PVI for 100 randomly chosen realizations for both the reference solution and the multiscale solution; relative overall saturation error and water-cut error; and histograms of the breakthrough times and cumulative oil production values depicted in Figure 5.32. Figure 5.32 demonstrates that there is generally a good match between the breakthrough time and cumulative oil production curves for the reference and multiscale solutions. However, we now observe that there is a slight bias in the multiscale results, for example, there is a small time-lag in the breakthrough times for the multiscale method. The bias can also be observed from the histograms in Figure 5.34, but the magnitude of the bias is small, and the multiscale solutions are generally quite close to the reference solution, as is illustrated in Figures 5.31 and 5.33.

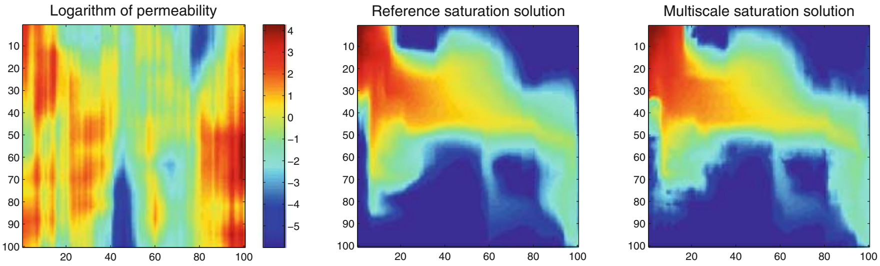


Fig. 5.31. An exponential variogram field with $l_1 = 0.5$, $l_2 = 0.1$, and $\sigma^2 = 2$, and a comparison of the reference saturation field at 0.6 PVI and the corresponding saturation field obtained using the stochastic multiscale method.

We now demonstrate that the bias in breakthrough time and cumulative oil production persists, but is efficiently reduced by increasing the number of realizations used to generate the multiscale basis functions. Figures 5.35, 5.36, and 5.37 show, respectively, the saturation and water-cut error for each of the 100 randomly selected realizations for the stochastic multiscale method with different sample sizes, the cumulative probability distribution of breakthrough times and cumulative oil production, and the corresponding histograms of the breakthrough times and the cumulative oil production values. Here, the sample size refers to the number of realizations selected in constructing multiscale basis functions. The plots show the following: the saturation and water-cut errors decay with increasing sample size; the time lag in the breakthrough times (also observed in the cumulative oil production) decays rapidly with increasing sample size, and that using 50 basis functions for each coarse-grid interface generates statistics that are nearly unbiased, and generally match the statistics derived from the set of reference solutions very well. Observe that a sample size of 50 gives rise to a linear system with 2025 unknowns, roughly 1/15 as many as in the fine-grid system.

Summary

In conclusion, we have developed and studied the stochastic mixed multiscale finite element method. This method solves stochastic porous media flow equation on the coarse grid using a set of pre-computed basis functions. The precomputed basis functions are constructed based on selected realizations of the stochastic permeability field, and thus span both spatial scales and uncertainties. The proposed method can be regarded as an extension of mixed MsFEM to stochastic porous media flow equations. The proposed approach does not require any interpolation in stochastic space and is capable of predicting the solution on the coarse grid. We present numerical results for two-phase immiscible flow in stochastic porous media which show that one can use few

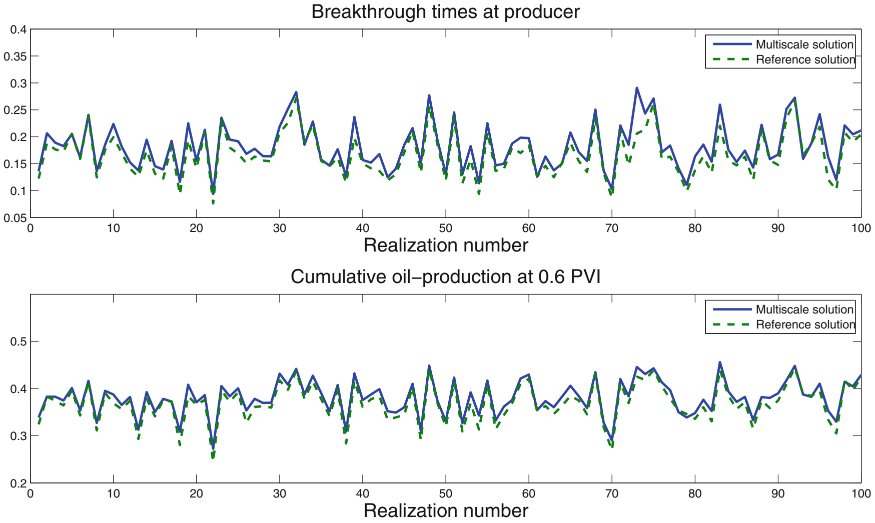


Fig. 5.32. Breakthrough time and cumulative oil production at 0.6 PVI for 100 random realizations from an exponential variogram field with $l_1 = 0.5$, $l_2 = 0.1$, and $\sigma^2 = 2$.

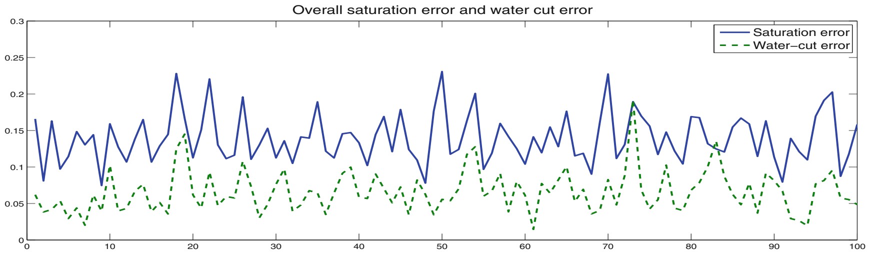


Fig. 5.33. L^2 errors of the saturation field and water-cut errors for 100 randomly chosen exponential variogram fields with $l_1 = 0.5$, $l_2 = 0.1$, and $\sigma^2 = 2$.

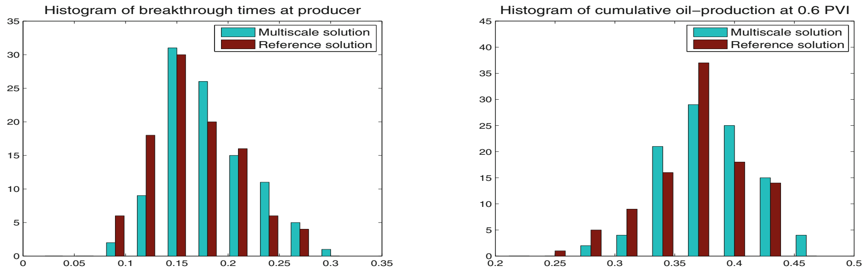


Fig. 5.34. Histograms of the breakthrough times and cumulative oil production values shown in Figure 5.32.

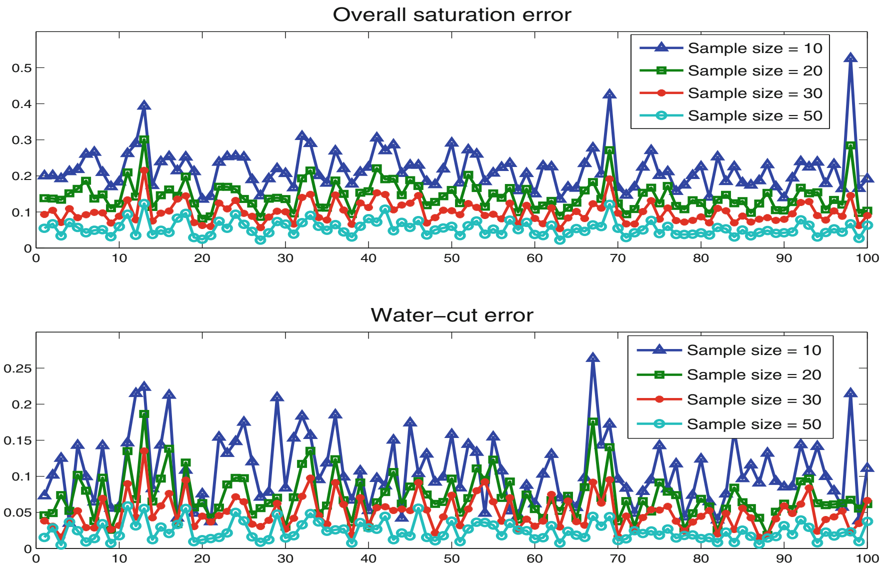


Fig. 5.35. Saturation and water-cut errors for solutions obtained using different number of permeability realizations to generate the multiscale basis functions.

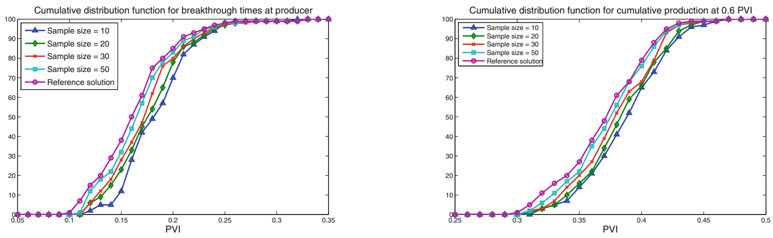


Fig. 5.36. Cumulative probability distribution for breakthrough time and cumulative oil production at 0.6 PVI.

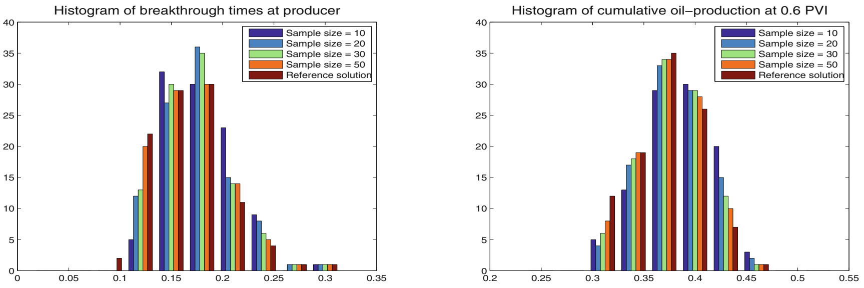


Fig. 5.37. Histograms of the breakthrough times and cumulative oil production.

basis functions in approximating the solutions of permeability fields with large uncertainty space. Finally, we would like to note that the proposed approaches can be easily combined with interpolation-based approaches in order to achieve greater flexibility.

5.7.2 The applications of MsFEMs to uncertainty quantification in inverse problems

In a number of papers [105, 89, 109, 88], applications of MsFEMs or upscaling methods to uncertainty quantification in inverse problems are discussed. The problem under consideration consists of finding stochastic realizations of the conductivity (or permeability) field given the measurement data and measurement errors (e.g., fractional flow (oil-cut) measurements defined by (2.43)). From the probabilistic point of view, this problem can be regarded as the conditioning of the permeability field to the measured data with associated measurement errors. Consequently, our goal is to sample from the conditional distribution $P(k|D)$, where k is the fine-scale permeability field and D is the measured data. Using the Bayes' theorem (e.g., [237]) we can write $\pi(k) = P(k|D) \propto P(D|k)P(k)$.

The techniques based on Metropolis–Hastings Markov chain Monte Carlo (MCMC) (see [237]) provide a rigorous framework for sampling the probability distribution $\pi(k)$ and obtaining the realizations of the conductivity field given measurements, albeit at high computational cost. The main idea of MCMC is to generate a Markov chain with $\pi(k)$ as its stationary distribution. A key step to this approach is to construct the desired transition probability distribution for the Markov chain. In Metropolis–Hastings MCMC, permeability samples, $k_1, ..., k_n, ...$ are generated. In particular, at k_n, a proposal k is generated using instrumental probability distribution $q(k|k_n)$. Furthermore, k is accepted as a sample with probability

$$p(k_n, k) = \min\left(1, \frac{q(k_n|k)\pi(k)}{q(k|k_n)\pi(k_n)}\right);$$

that is, take $k_{n+1} = k$ with probability $p(k_n, k)$, and $k_{n+1} = k_n$ with probability $1 - p(k_n, k)$.

In the Metropolis–Hastings MCMC algorithm, the major computational cost is to compute the value of the target distribution $\pi(k)$, which involves solving the coupled nonlinear PDE system (5.1) and (5.2) on the fine grid. Generally, the Metropolis–Hastings MCMC method requires many iterations before it converges to the steady state. To assess the uncertainty accurately, one needs to generate a large number of different samples. Thus, the direct (full) MCMC simulations are usually prohibitively expensive. Moreover, the acceptance rate of the direct MCMC method can be very low, due to the large dimensions of the permeability field. As a result, most of the CPU time is spent on rejected samples.

An important way to improve the direct MCMC method is to increase the acceptance rate by modifying the proposal distribution $q(k|k_n)$. Typically, some simplified models can be used to do so (e.g., [66, 179]). In [105, 89, 109], we discuss algorithms that use approximate and inexpensive coarse-scale simulations based on MsFEMs to speedup MCMC calculations. In particular, we consider an application to two-phase flow and transport simulations where the pressure equation is upscaled using the MsFVEM or mixed MsFEM or stochastic mixed MsFEM, and the saturation equation is upscaled using a simple volume averaging

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla f(\overline{S}) = 0. \tag{5.57}$$

Although, this type of upscaling can introduce some errors (see Figure 2.11), it can be used in uncertainty quantification in inverse problems for the following reasons. First, this approach, which combines MsFEMs for the pressure equation and primitive upscaled model for the saturation equation, is very inexpensive. Second, we have observed that there is a strong correlation between the misfit corresponding to fine- and coarse-scale fractional flows.

Denote by D_k^* the coarse-scale data computed using MsFEMs. In the applications to two-phase flow and transport, MsFEMs are used for the pressure equation with permeability k and (5.57) for the saturation equation. Furthermore, we denote by $\pi^*(k) = P(k|D^*)$ the corresponding coarse-scale approximation of the target distribution $\pi(k)$. In general, one can perform offline simulations to estimate a statistical relation between the coarse-scale output D_k^* and the fine-scale output D_k via offline simulations for different ks sampled from the prior distribution. Based on this relation, π^* can be estimated (see [109]). Our main emphasis is the use of physics-based coarse-scale models for uncertainty quantification in inverse problems.

Using the coarse-scale distribution $\pi^*(k)$ as a filter, the preconditioned MCMC was proposed in [105]. In this approach, the coarse-scale simulation is used in the second stage to screen the proposal before running a fine-scale simulation. More precisely, after making a proposal as in Metropolis–Hastings MCMC, the coarse-scale simulation is performed and the proposal is screened using π^* distribution. If the proposal is accepted at this stage, only then a fine-scale simulation is performed for the proposed permeability field to decide whether to accept the proposal. Because the computation of the coarse-scale solution is very cheap, this step can be implemented very quickly to decide whether to run fine-scale simulations. The second step of the algorithm serves as a filter that avoids unnecessary fine-scale runs for the rejected samples. In [105], we show that the modified Markov chain is ergodic and converges to the correct distribution. We present numerical results for permeability fields generated using two-point correlation functions (see (5.50)) in [105]. Our results demonstrate that preconditioned MCMC has similar convergence properties, it has higher acceptance rates, and provides an order of magnitude of CPU saving. We refer to [105] for details.

An important type of proposal distribution can be derived from the Langevin diffusion, as proposed by Grenander and Miller [138], which uses the gradient of the posterior in the proposal. The use of the gradient information in inverse problems for subsurface characterization is not new (e.g., see [216]). The use of gradient information allows us to achieve high acceptance rates (e.g., [249]). In [89], we proposed the preconditioned coarse-gradient Langevin algorithm, where the gradient information based on π^* was used for generating a proposal. This step is much cheaper than the corresponding step involving a fine-scale gradient of π because the simulations are performed on the coarse grid. Furthermore, this proposal is screened using coarse-scale models as in preconditioned MCMC discussed above. The details of this algorithm can be found in [89], where we presented numerical results. Numerical results show that preconditioned coarse-gradient Langevin algorithms are efficient and can give similar performance as the fine-scale Langevin algorithms with much less computational cost. We refer to [89] for details.

The MCMC method used in these simulations employs either the mixed MsFEM or MsFVEM in the preconditioning step. If a proposal is accepted by the preconditioning step, the proposed algorithms compute the fine-scale solutions corresponding to the proposed permeability field. At this stage, we have already precomputed basis functions that can be further used to reconstruct the velocity field on the fine scale. Then the transport equation can be solved on the fine grid coupled with the coarse-grid pressure equation. This approach provides an accurate approximation to the production data on the fine grid as discussed earlier and can be used to replace the fine-scale computation in the last stage. In this procedure, the basis functions are not updated in time, or updated only in a few coarse blocks. Thus the fine-scale computation in the last stage of MCMC algorithms can also be implemented quickly. Because the basis functions from the first-stage is re-used for the fine-scale computation, this combined multiscale approach can be very efficient for our sampling problem.

For problems involving a very high dimensional uncertainty space, such as permeability fields described by the exponential variogram, it is often advantageous to use an approximate response surface in computing Langevin proposals. We proposed the use of sparse interpolation techniques based on coarse-scale models in obtaining the approximation of the response surface in [88]. In this case, the posterior distribution is interpolated using sparse interpolation techniques. We first compute the posterior distribution at sparse locations that correspond to some selected realizations of the permeability field. These computations are performed on the coarse grid as before with MsFVEM and thus they are inexpensive. Furthermore, the posterior distribution is approximated using polynomial interpolation. Based on the interpolated posterior distribution, Langevin samples are proposed using analytical gradients of the posterior distribution. The numerical simulations show that one can achieve further gains in CPU if interpolation is used. We refer to [88] for further details.

We note that there are other efficient approaches (e.g., [275, 185, 128, 129]) which are used in uncertainty quantification in inverse problems for porous media flows. Here, our goal was simply to discuss an application of MsFEM to porous media flows within MCMC methods.

5.8 Discussions

In this chapter, we discussed the applications of MsFEMs to porous media flow and transport in the context of two-phase immiscible flow and transport. In a number of recent findings, the latter has been extended to more complicated porous media equations involving compressibility, gravity, and three phases as demonstrated in Sections 5.5 and 5.6. In general, MsFEMs offer a great advantage when the heterogeneities do not change significantly or these changes can be localized. This allows us to solve the flow equations on a coarse grid. In a more complex situation, this may not be the case and one has to be careful applying multiscale methods.

Another interesting application of multiscale finite element methods is to inverse problems. In [246], the authors took advantage of the adaptivity of multiscale methods to speedup inverse problems associated with finding permeability fields given average flow rates at the well and some other prior information. During the inversion procedure, the permeability is updated only in local regions using time travel inversion. Because of local changes in the permeability heterogeneities, multiscale basis functions are constructed only in a few coarse blocks and the solution is rapidly computed. This leads to very fast inversion and the CPU time for finding appropriate permeability samples defined on a multi million grid block is very small (less than two minutes on a PC).