2

Multiscale finite element methods for linear problems and overview

2.1 Summary

In this section, the main concept of multiscale finite element methods (MsFEM) is presented. We keep the presentation simple to make it accessible to a broader audience. Two main ingredients of MsFEMs are the global formulation of the method and the construction of basis functions. We discuss global formulations using various finite element, finite volume, and mixed finite element methods. As for multiscale basis functions, the subgrid capturing errors are discussed. We present simplified computations of basis functions for cases with scale separation. We also discuss the improvement of subgrid capturing errors via oversampling techniques. Finally, we present some representative numerical examples and discuss the computational cost of MsFEMs. Analysis of some representative cases is presented in Chapter 6.

2.2 Introduction to multiscale finite element methods

We start our discussion with the MsFEM for linear elliptic equations

$$Lp = f \text{ in } \Omega, \tag{2.1}$$

where Ω is a domain in $\mathbb{R}^d$ $(d = 2, 3)$, $Lp := -\text{div}(k(x)\nabla p)$, and $k(x)$ is a heterogeneous field varying over multiple scales. We note that MsFEM can be easily extended to systems such as elasticity equations, as well as to nonlinear problems (see Section 2.4 and Chapter 3). The choice of the notations $k(x)$ and $p(x)$ in (2.1) is used because of the applications of the method to porous media flows later in the book. We note that the tensor $k(x) = (k_{ij}(x))$ is assumed to be symmetric and satisfies $\alpha|\xi|^2 \leq k_{ij}\xi_i\xi_j \leq \beta|\xi|^2$, for all $\xi \in \mathbb{R}^d$ and with $0 < \alpha < \beta$. We omit x dependence when there is no ambiguity and assume the summation over repeated indices (Einstein summation convention) unless otherwise stated.

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_2,

MsFEMs consist of two major ingredients: multiscale basis functions and a global numerical formulation that couples these multiscale basis functions. Basis functions are designed to capture the multiscale features of the solution. Important multiscale features of the solution are incorporated into these localized basis functions which contain information about the scales that are smaller (as well as larger) than the local numerical scale defined by the basis functions. A global formulation couples these basis functions to provide an accurate approximation of the solution. Next, we discuss some basic choices for multiscale basis functions and global formulations.

Basis functions. First, we discuss the basis function construction. Let $\mathcal{T}_h$ be a usual partition of Ω into finite elements (triangles, quadrilaterals, and so on). We call this partition the coarse grid and assume that the coarse grid can be resolved via a finer resolution called the fine grid. For clarity of this exposition, we plot rectangular coarse and fine grids in Figure 2.1 (left figure). Let x_i be the interior nodes of the mesh $\mathcal{T}_h$ and ϕ_i^0 be the nodal basis of the standard finite element space $W_h = \text{span}\{\phi_i^0\}$. For simplicity, one can assume that W_h consists of piecewise linear functions if $\mathcal{T}_h$ is a triangular partition. Denote by $S_i = \text{supp}(\phi_i^0)$ (the support of ϕ_i^0) and define ϕ_i with support in S_i as follows

$$L\phi_i = 0 \text{ in } K, \quad \phi_i = \phi_i^0 \text{ on } \partial K, \quad \forall K \in \mathcal{T}_h, \;\; K \subset S_i; \tag{2.2}$$

that is multiscale basis functions coincide with standard finite element basis functions on the boundaries of a coarse-grid block K, and are oscillatory in the interior of each coarse-grid block. Throughout, K denotes a coarse-grid block. Note that even though the choice of ϕ_i^0 can be quite arbitrary, our main assumption is that the basis functions satisfy the leading-order homogeneous equations when the right-hand side f is a smooth function (e.g., L^2 integrable). We would like to remark that MsFEM formulation allows one to take advantage of scale separation which is discussed later in the book. In particular, K can be chosen to be a domain smaller than the coarse grid as illustrated in Figure 2.1 (right figure) if the small region can be used to represent the heterogeneities within the coarse-grid block. In this case, the basis function has the formulation (2.2), except K is replaced by a smaller region, K_{loc}, $L(\phi_i) = 0$ in K_{loc}, $\phi_i = \phi_i^0$ on ∂K_{loc}, where the values of ϕ_i^0 inside K are used in imposing boundary conditions on ∂K_{loc}. In general, one solves (2.2) on the fine grid to compute basis functions. In some cases, the computations of basis functions can be performed analytically. To illustrate the basis functions, we depict them in Figure 2.2. On the left, the basis function is constructed when K is a coarse partition element, and on the right, the basis function is constructed by taking K to be an element smaller than the coarse-grid block size. Note that a bilinear function in Figure 2.2 (right figure) is used to demonstrate boundary conditions on a small computational domain and this bilinear function is not a part of the basis function. In this case, the assembly of the stiffness matrix uses only the information in small computational regions and the basis function can be "periodically" extended

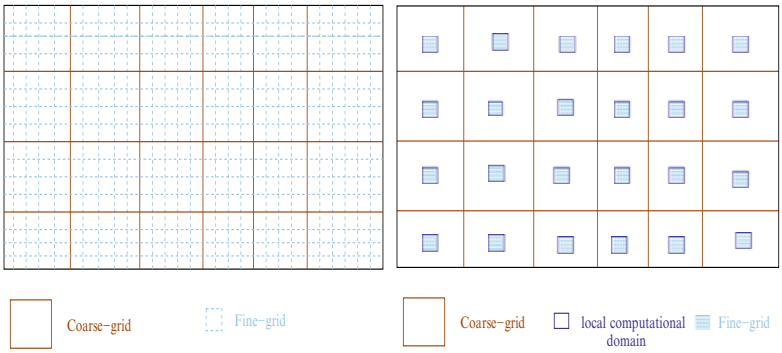

Fig. 2.1. Schematic description of a coarse grid.

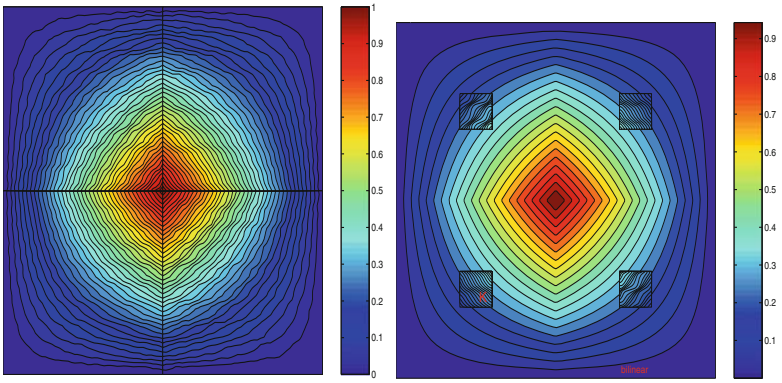

Fig. 2.2. Example of basis functions. Left: basis function with K being a coarse element. Right: basis function with K being RVE (bilinear function demonstrates only the boundary conditions on RVE and is not a part of the basis function).

to the coarse-grid block, if needed (see later discussions and Section 2.6). Computational regions smaller than the coarse-grid block are used if one can use smaller regions to characterize the local heterogeneities within the coarse grid block (e.g., periodic heterogeneities). We call such regions Representative Volume Element (RVE) following standard practice in engineering. More precisely, we assume that the size of the RVE is much larger than the characteristic length scale. In this case, one can use the solution in RVE with prescribed boundary conditions to represent the solution in the coarse block as is done in homogenization (e.g., [43, 164]). Later, we briefly discuss an extension of the method to problems with singular right-hand sides. In this case, it is necessary to include basis functions with singular right-hand sides. Once the basis functions are constructed, we denote by $\mathcal{P}_h$ the finite element space spanned by ϕ_i

$$\mathcal{P}_h = \text{span}\{\phi_i\}.$$

Global formulation. Next, we discuss the global formulation of MsFEM. The representation of the fine-scale solution via multiscale basis functions allows reducing the dimension of the computation. When the approximation of the solution $p_h = \sum_i p_i \phi_i(x)$ (p_i are the approximate values of the solution at coarse-grid nodal points) is substituted into the fine-scale equation, the resulting system is projected onto the coarse-dimensional space to find p_i. This can be done by multiplying the resulting fine-scale equation with coarse-scale test functions. Other approaches can be taken for general nonlinear problems. In the case of Galerkin finite element methods, when the basis functions are conforming ($\mathcal{P}_h \subset H_0^1(\Omega)$), the MsFEM is to find $p_h \in \mathcal{P}_h$ such that

$$\sum_K \int_K k \nabla p_h \cdot \nabla v_h dx = \int_\Omega f v_h dx, \quad \forall\ v_h \in \mathcal{P}_h. \tag{2.3}$$

One can choose the test functions from W_h (instead of $\mathcal{P}_h$) and arrive at the Petrov–Galerkin version of the MsFEM as introduced in [143]. Find $p_h \in \mathcal{P}_h$ such that

$$\sum_K \int_K k \nabla p_h \cdot \nabla v_h dx = \int_\Omega f v_h dx, \quad \forall\ v_h \in W_h. \tag{2.4}$$

We note that in both formulations (2.3) and (2.4), the fine-scale system is multiplied by coarse-scale test functions and, thus, the resulting system is coarse-dimensional.

Equation (2.3) or (2.4) couples the multiscale basis functions. This gives rise to a linear system of equations for finding the values of the solution at the nodes of the coarse-grid block, thus, the resulting system of linear equations determines the solution on the coarse grid. To show this, for simplicity, we consider the Petrov–Galerkin formulation of the MsFEM (see (2.4)). Representing the solution in terms of multiscale basis functions, $p_h = \sum_i p_i \phi_i$, it is easy to show that (2.4) is equivalent to the following linear system,

$$A p_{\text{nodal}} = b, \tag{2.5}$$

where $A = (a_{ij})$, $a_{ij} = \sum_K \int_K k \nabla \phi_i \nabla \phi_j^0 dx$. $p_{\text{nodal}} = (p_1, ..., p_i, ...)$ are the nodal values of the coarse-scale solution, and $b = (b_i)$, $b_i = \int_\Omega f \phi_i^0 dx$. Here, we do not consider the discretization of boundary conditions. As in the case of standard finite element methods, the stiffness matrix A has sparse structure. We note that the computation of the stiffness matrix requires the integral computation for a_{ij} and b_i. The computation of a_{ij} requires the evaluation of the integrals on the fine grid. One can use a simple quadrature rule, for example, one point per fine grid cell. In this case, $\int_K k \nabla \phi_i \nabla \phi_j^0 dx \approx \sum_{\tau \subset K} (k \nabla \phi_i)|_\tau \nabla \phi_j^0$, where τ denotes a fine grid block and $(k \nabla \phi_i)|_\tau$ is the value of the flux within a fine grid block τ. Note that when source terms or mobilities change, one can pre-compute the stiffness matrix once and re-use it. For example, if the source terms change, the stiffness matrix will remain the same and one needs to re-compute b_i. If mobilities ($\lambda(x)$

in (1.2)) change and remain a smooth function, one can modify a_{ij} using a piecewise constant approximation for $\lambda(x)$. In this case, the modified stiffness matrix elements a_{ij}^{λ} have the form $a_{ij}^{\lambda} \approx \sum_K \lambda_K \int_K k\nabla\phi_i \nabla\phi_j^0 dx$, where a_{ij}^{λ} are the elements of the stiffness matrix corresponding to (1.2), λ_K are approximate values of $\lambda(x)$ in K, and the integrals $\int_K k\nabla\phi_i\nabla\phi_j^0 dx$ are pre-computed. Later on, we derive some explicit expressions for the elements of the stiffness matrix in the one-dimensional case.

If the local computational domain is chosen to be smaller than the coarse-grid block, then one can use an approximation of the basis functions in RVE (local domain) to represent the left-hand side of (2.4) (or (2.3)). We assume that the information within RVE can be used to characterize the local solution within the coarse-grid block such that

$$\frac{1}{|K|}\int_K k\nabla\phi_i dx \approx \frac{1}{|K_{\text{loc}}|}\int_{K_{\text{loc}}} k\nabla\tilde{\phi}_i dx, \tag{2.6}$$

where K_{loc} refers to local computational region (RVE) and $\tilde{\phi}_i$ is the basis function defined in K_{loc} and given by the solution of $\text{div}(k\nabla\tilde{\phi}_i) = 0$ in K_{loc} with boundary conditions $\tilde{\phi}_i = \phi_i^0$ on ∂K_{loc}. Equation (2.6) holds, for example, in the general G-convergence setting where homogenization by periodization (also called the principle of periodic localization) can be performed (see [164]) and the size of the RVE is assumed to be much larger than the characteristic length scale. One can approximate the left-hand side of (2.4) based on RVE computations. In particular, the elements of the stiffness matrix (see (2.5)) $a_{ij} = \sum_K \int_K k\nabla\phi_i\nabla\phi_j^0 dx$ can be approximated using

$$\frac{1}{|K|}\int_K k\nabla\phi_i\nabla\phi_j^0 dx \approx \frac{1}{|K_{\text{loc}}|}\int_{K_{\text{loc}}} k\nabla\tilde{\phi}_i\nabla\phi_j^0 dx.$$

A similar approximation can be done for (2.3). In the general G-convergence setting, this approximation holds in the limit $\lim_{h\to 0}\lim_{\epsilon\to 0}$ (see Section 2.6 for details), and for periodic problems, one can justify this approximation in the limit $\lim_{\epsilon/h\to 0}$. In periodic problems, one can also take advantage of two-scale homogenization expansion and this is discussed in Section 2.6 along with further discussions on the use of smaller regions. Similar approximation can be done for the right-hand side of (2.4) (or (2.3)).

As we discussed earlier, using multiscale basis functions, a fine-scale approximation of the solution can be easily computed. In particular, $p_h = \sum_i p_i\phi_i$ provides an approximation of the solution, where p_i are the values of the solution at the coarse nodes obtained via (2.5). When regions smaller than the coarse-grid block are used for computing basis functions, $p_h = \sum_i p_i\phi_i$ provides approximate fine-scale details of the solution only in RVE regions. One can use the periodic homogenization concept to extend the fine-scale features in RVE to the entire domain. This is discussed in Section 2.6.

In the above discussion, we presented the simplest basis function construction and a global formulation. In general, the global formulation can be easily

modified and various global formulations based on finite volume, mixed finite element, discontinuous Galerkin finite element, and other methods can be derived. Many of them are studied in the literature and some of them are discussed here.

As for basis functions, the choice of boundary conditions in defining the multiscale basis functions plays a crucial role in approximating the multiscale solution. Intuitively, the boundary condition for the multiscale basis function should reflect the multiscale oscillation of the solution p across the boundary of the coarse grid element. By choosing a linear boundary condition for the basis function, we create a mismatch between the exact solution p and the finite element approximation across the element boundary. In Section 2.3, we discuss this issue further and introduce a technique to alleviate this difficulty. We would like to note that in the one-dimensional case this issue is not present because the boundaries of the coarse element consist of isolated points.

The MsFEM can be naturally extended to solve nonlinear partial differential equations. As in the case of linear problems, the main idea of MsFEM remains the same with the exception of basis function construction. Because of nonlinearities, the multiscale basis functions are replaced by multiscale maps, which are in general nonlinear maps from W_h to heterogeneous fields (see Chapter 3).

Pseudo-code. MsFEM can be implemented within an existing finite element code. Below, we present a simple pseudo-code that outlines the implementation of MsFEM. Here, we do not discuss coarse-grid generation. We note that the latter is important for the accuracy, robustness, and efficient parallelization of MsFEM and is briefly discussed in Section 2.9.3.

Algorithm 2.2.1

Set coarse mesh configuration from fine-scale mesh information.
For each coarse grid block n do
- For each vertex i
- Solve for ϕ_n^i satisfying $-L(\phi_n^i) = 0$ and boundary conditions (see (2.2))
- End for.

End do
Assemble stiffness matrix on the coarse mesh (see (2.5), also (2.3) or (2.4)).
Assemble the external force on the coarse mesh (see (2.5), also (2.3) or (2.4)).
Solve the coarse formulation.

Comments on the assembly of stiffness matrix. One can use the representation of multiscale basis functions via fine-scale basis functions to assemble the stiffness matrix. This is particularly useful in code development. Assume that multiscale basis function (in discrete form) ϕ_i can be written as

$$\phi_i = d_{ij}\phi_j^{0,f},$$

where $D = (d_{ij})$ is a matrix and $\phi_j^{0,f}$ are fine-scale finite element basis functions (e.g., piecewise linear functions). The ith row of this matrix contains the fine-scale representation of the ith multiscale basis function. Substituting this expression into the formula for the stiffness matrix a_{ij} in (2.3), we have

$$a_{ij} = \int_\Omega k\nabla\phi_i\nabla\phi_j dx = d_{il}\int_\Omega k\nabla\phi_l^{0,f}\nabla\phi_m^{0,f}dx\ d_{jm}.$$

Denoting the stiffness matrix for the fine-scale problem by $A^f = (a_{lm}^f)$, $a_{lm}^f = \int_\Omega k\nabla\phi_l^{0,f}\nabla\phi_m^{0,f}dx$ we have

$$A = DA^fD^T.$$

Similarly, for the right-hand side, we have $b = \int_\Omega \phi_i f dx = Db^f$, where $b^f = (b_i^f)$, $b_i^f = \int_\Omega f\phi_i^{0,f}dx$. This simplification can be used in the assembly of the stiffness matrix. The similar procedure can be done for the Petrov–Galerkin MsFEM (see (2.4)).

One-dimensional example. In one-dimensional case, the basis functions and the stiffness matrix (see (2.5)) can be computed almost explicitly. For simplicity, we consider

$$-(k(x)p')' = f,$$

$p(0) = p(1) = 0$, where $'$ refers to the spatial derivative. We assume that the interval $[0,1]$ is divided into N segments $0 = x_0 < x_1 < x_2 < \cdots < x_i < x_{i+1} < \cdots < x_N = 1$. The multiscale basis function for the node i is given by

$$(k(x)\phi_i')' = 0 \tag{2.7}$$

with the support in $[x_{i-1}, x_{i+1}]$. In the interval $[x_{i-1}, x_i]$, the boundary conditions for the basis function ϕ_i are defined as $\phi_i(x_{i-1}) = 0$, $\phi_i(x_i) = 1$. In the interval $[x_i, x_{i+1}]$, the boundary conditions for the basis function ϕ_i are defined as $\phi_i(x_i) = 1$, $\phi_i(x_{i+1}) = 0$. Note that for the computation of the elements of the stiffness matrix, we do not need an explicit expression of ϕ_i and instead, we simply need to compute $k(x)\phi_i'$. From (2.7), it is easy to see that $k(x)\phi_i' = \text{const}$, where the constants are different in $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$. This constant can be easily computed by writing $\phi_i' = \text{const}/k(x)$ and integrating it over $[x_{i-1}, x_i]$. This yields

$$k(x)\phi_i' = \frac{1}{\int_{x_{i-1}}^{x_i}\frac{dx}{k(x)}}$$

on $[x_{i-1}, x_i]$ and

$$k(x)\phi_i' = -\frac{1}{\int_{x_i}^{x_{i+1}}\frac{dx}{k(x)}}$$

on $[x_i, x_{i+1}]$. Then, the elements of the stiffness matrix A (see (2.4)) are given by

$$\begin{aligned} a_{ij} &= \int_{x_{i-1}}^{x_i} k(x)\phi_i^{'}(\phi_j^0)^{'}dx + \int_{x_i}^{x_{i+1}} k(x)\phi_i^{'}(\phi_j^0)^{'}dx \\ &= \frac{1}{\int_{x_{i-1}}^{x_i}\frac{dx}{k(x)}}\int_{x_{i-1}}^{x_i}(\phi_j^0)^{'}dx - \frac{1}{\int_{x_i}^{x_{i+1}}\frac{dx}{k(x)}}\int_{x_{i-1}}^{x_i}(\phi_j^0)^{'}dx. \end{aligned} \tag{2.8}$$

Taking into account that $\int_{x_{i-1}}^{x_i}(\phi_{i-1}^0)^{'}dx = -1$, $\int_{x_{i-1}}^{x_i}(\phi_i^0)^{'}dx = 1$, $\int_{x_i}^{x_{i+1}}(\phi_i^0)^{'}dx = -1$, $\int_{x_i}^{x_{i+1}}(\phi_{i+1}^0)^{'}dx = 1$, we have

$$a_{i,i-1} = -\frac{1}{\int_{x_{i-1}}^{x_i}\frac{dx}{k(x)}}, \quad a_{ii} = \frac{1}{\int_{x_{i-1}}^{x_i}\frac{dx}{k(x)}} + \frac{1}{\int_{x_i}^{x_{i+1}}\frac{dx}{k(x)}}, \quad a_{i,i+1} = -\frac{1}{\int_{x_i}^{x_{i+1}}\frac{dx}{k(x)}}.$$

Consequently, the stiffness matrix has a tridiagonal form and the linear system is (2.5), where $b_i = \int_0^1 f\phi_i^0 dx$.

In Figure 2.3, we illustrate the solution and a few multiscale basis functions. We refer to [147] for the analysis in the one-dimensional case.

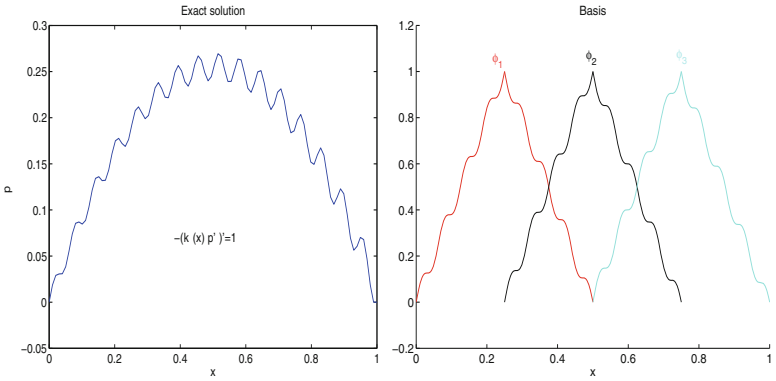

Fig. 2.3. An illustration of one-dimensional basis functions and the solution.

2.3 Reducing boundary effects

2.3.1 Motivation

The boundary conditions for the basis functions play a crucial role in capturing small-scale information. If the local boundary conditions for the basis functions do not reflect the nature of the underlying heterogeneities, MsFEMs can have large errors. These errors result from the resonance between the

coarse-grid size and the characteristic length scale of the problem. When the coefficient $k(x)$ is a periodic function varying over the ϵ scale ($k(x) = k(x/\epsilon)$), the convergence rate of MsFEM contains a term ϵ/h (see [147]), which is large when $h \approx \epsilon$. Recall that h is the coarse mesh size. As illustrated by the error analysis of [147], the error due to the resonance manifests as a ratio between the wavelength of the small-scale oscillation and the grid size; the error becomes large when the two scales are close. A deeper analysis based on the homogenization theory shows the main source of the resonance effect. By a judicious choice of boundary conditions for basis functions, we can reduce the resonance errors significantly. Some approaches including the use of reduced problems based on the solutions of one-dimensional problems along the boundaries (e.g., [145, 159, 160]) and oversampling methods (e.g., [145, 107, 73]) are studied in the literature with the goal of reducing resonance errors. In general, one can construct multiscale basis functions in various different ways (see, e.g., [266, 273] for energy minimizing basis functions). Here, we focus on oversampling methods.

Next, we present an outline of the analysis that motivates the oversampling method. We consider a simple case with two distinct scales (i.e., $k(x) = k(x, x/\epsilon)$) and assume that k is a periodic function with respect to x/ϵ. In this case, the solution has a well-known multiscale expansion (see, e.g., [43, 164] or Appendix B)

$$p = p_0 + \epsilon\chi^j \frac{\partial}{\partial x_j} p_0 + \epsilon\theta_\epsilon^p,$$

where p_0 satisfies the homogenized equation $-\mathrm{div}(k^*(x)\nabla p_0) = f$. The homogenized coefficients are defined via an auxiliary (cell) problem over a period of size ϵ. To illustrate this, we denote the fast variable by $y = x/\epsilon$ and, thus, the coefficients have the form $k(x, y)$. Then, $k^*(x) = (1/|Y|)\int_Y k(x, y)(I + \nabla_y\chi(x, y))dy$, where $\chi = (\chi^1, ..., \chi^d)$ is a solution of

$$\mathrm{div}_y(k(x, y)(I + \nabla_y\chi(x, y))) = 0 \tag{2.9}$$

in the period Y for a fixed x (see [43, 164] for more details). For simplicity, one can assume that x represents a coarse grid block. If there is no slow dependence with respect to x in the coefficients, $k = k(x/\epsilon) = k(y)$, then there is only one cell problem (2.9) for the entire domain Ω. It can be shown that $p_0 + \epsilon\chi^j \partial p_0/\partial x_j$ approximates the solution p in H^1 norm for small ϵ (see [43, 164] or Appendix B for the details).

Following multiple scale expansion, as discussed above (see also Appendix B), we can write a similar expansion for the basis function

$$\phi_i = \phi_i^1 + \epsilon\theta_\epsilon, \tag{2.10}$$

where $\phi_i^1 = \phi_i^0 + \epsilon\chi^j \partial\phi_i^0/\partial x_j$ is the part of the basis function that has the same nature of oscillations near boundaries as the approximation of the fine-scale solution $p_0 + \epsilon\chi^j \partial p_0/\partial x_j$. Assuming ϕ_i^0 is a linear function, it can be

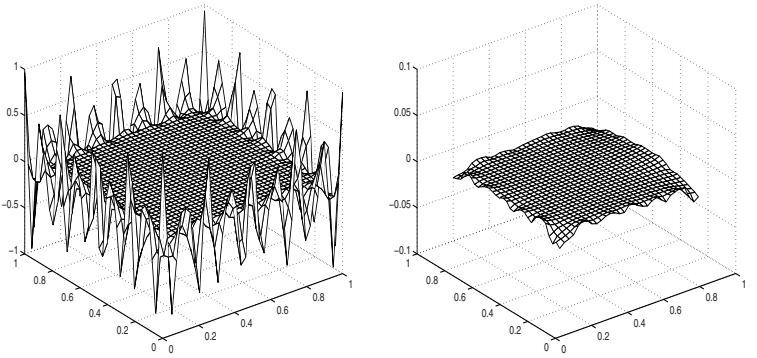

Fig. 2.4. An illustration of boundary layer function θ_ϵ. Left: θ_ϵ in the coarse element with oscillatory boundary conditions. Right: θ_ϵ in ϵ distance away from the boundaries.

easily shown that θ_ϵ satisfies $\mathrm{div}(k\nabla\theta_\epsilon) = 0$ in K and $\theta_\epsilon = -\chi^j \partial\phi_i^0/\partial x_j$ on ∂K. If one can ignore $\epsilon\theta_\epsilon$ in (2.10), then MsFEM will converge independently of the resonance error. The term $\epsilon\theta_\epsilon$ is due to the mismatch between the fine-scale solution and multiscale finite element solution along the boundaries of the coarse-grid block where the multiscale finite element solution is linear. This mismatch error propagates into the interior of the coarse-grid block. The analysis shows that the MsFEM error is dominated by θ_ϵ. In Figure 2.4, we depict $\theta_\epsilon(x)$ and the same $\theta_\epsilon(x)$ which is ϵ distance away from the boundaries. It is clear from this figure that the oscillations decay quickly as we move away from the boundaries. To avoid these oscillations, one needs to sample a larger domain and use only interior information to construct the basis functions. The decay of these oscillations basically dictates how large the sampling region should be chosen.

2.3.2 Oversampling technique

Motivated by the above discussion and the convergence analysis of [147], Hou and Wu proposed an oversampling method in [145] to overcome the difficulty due to scale resonance. Because the boundary layer in the first-order corrector is thin, we can sample in a domain with the size larger than h and use only the interior sampled information to construct the basis functions. By doing this, we can reduce the influence of the boundary layer in the larger sample domain on the basis functions significantly. It is intuitively clear from Figure 2.4 that the effects of artificial boundary conditions are significantly reduced for this special two-scale example.

Specifically, let ϕ_j^E be the basis functions satisfying the homogeneous elliptic equation in the larger domain $K_E \supset K$ (see Figure 2.5). We then form the actual basis ϕ_i by linear combination of ϕ_j^E,

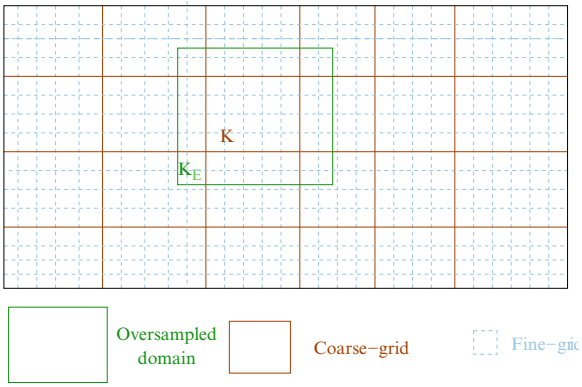

Fig. 2.5. Schematic description of oversampled region.

$$\phi_i = \sum_{j=1}^{d} c_{ij} \phi_j^E \ .$$

The coefficients c_{ij} are determined by condition $\phi_i(x_j) = \delta_{ij}$, where x_j are nodal points. Extensive numerical experiments have demonstrated that the oversampling technique does improve the numerical error substantially in many applications. Some numerical examples are presented in Section 2.9. On the other hand, the oversampling technique results in a nonconforming MsFEM method, where the basis functions are discontinuous along the edges of coarse-grid blocks. In [107] we perform a careful estimate of the nonconforming errors. The analysis shows that the nonconforming error is indeed small, and consistent with our numerical results [145, 146]. Our analysis also reveals another source of resonance, which is the mismatch between the mesh size and the "perfect" sample size. In the case of a periodic structure, the "perfect" sample size is the length of an integer multiple of the period. We call the new resonance the "cell resonance". In the error expansion, this resonance effect appears as a higher-order correction. In numerical computations, we found that the cell resonance error is generally small, and is rarely observed in practice. Nonetheless, it is possible to completely eliminate this cell resonance error by using the oversampling technique to construct the basis functions, but using piecewise linear functions as test functions. This reduces the resonance error further (see [143]).

2.4 Generalization of MsFEM: A look forward

Next, we present a general framework of MsFEMs (following Efendiev, Hou, and Ginting [104]) which is further discussed in Chapter 3. Consider

$$Lp = f, \tag{2.11}$$

where $L : X \to Y$ is an operator. The objective of the MsFEM is to approximate p on the coarse grid. Denote by W_h a family of finite-dimensional space such that it possesses an approximation property (see [274], [229]), as before. Here h, as before, is a scale of computation (coarse grid). In general, multiscale basis functions are replaced by multiscale maps defined as $E^{MsFEM} : W_h \to \mathcal{P}_h$. For each element $v_h \in W_h$, $v_{r,h} = E^{MsFEM} v_h$ is defined as

$$L^{\text{map}} v_{r,h} = 0 \text{ in } K, \tag{2.12}$$

where L^{map} can be, in general, different from L (e.g., can be a discrete operator). Note that v_h (the quantity with the subscript h) denotes the coarse-scale approximation and $v_{r,h}$ (the quantity with the subscript r, h) denotes the fine-scale approximation. For linear problems, we simply used the subscript h to denote fine-scale approximations.

Note that L^{map} allows us to capture the effects of the small scales. Moreover, the domains different from the target coarse block K can be used in the computations of the local solutions. To solve (2.12) one needs to impose boundary and initial conditions. This issue needs to be resolved on a case-by-case basis, and the main idea is to interpolate v_h onto the underlying fine grid.

To find a solution of (2.11) in $\mathcal{P}_h$, one can substitute p_h (which denotes a coarse-scale solution defined in W_h) into (2.11) discretized on the fine grid. Because p_h is defined on the coarse grid, the resulting system is projected onto the coarse-dimensional space. This can be done in various ways. A common approach is to multiply the resulting fine-scale system by coarse-scale test functions; that is find $p_h \in W_h$ (consequently $p_{r,h} \in \mathcal{P}_h$) such that

$$\langle L^{\text{global}} p_{r,h}, v_h \rangle = \langle f, v_h \rangle, \quad \forall v_h \in W_h, \tag{2.13}$$

where $\langle \cdot, \cdot \rangle$ denotes a duality between X and Y (defined for the discrete variational formulation), and L^{global} can be, in general, different from L. We note that the fine-scale system $L^{\text{global}} p_{r,h} - f$ is multiplied by coarse-scale test functions. One can also minimize the residual $L^{\text{global}} p_{r,h} - f$ at some nodes to obtain a coarse-dimensional problem. Other approaches based on upscaled equations can also be used (see Section 5.4). In general, L^{map} and L^{global} can be different for nonlinear problems. Moreover, p_h can represent only some of the physical variables involved in the simulations (see Section 5.4).

The convergence of the MsFEM is to show that $p_h \approx p^*$ and $p_{r,h} \approx p$ in appropriate spaces for small h, where $p_{r,h} = E^{MsFEM} p_h$. Here p^* is a coarse-scale solution of (2.11). The correct choices of L^{map} and L^{global} are the essential part of MsFEM and guarantee the convergence of the method. We note that for linear elliptic equations, L^{map} is a linear map, and consequently, $\mathcal{P}_h$ is a linear space spanned by $E^{MsFEM} \phi_j^0$, where $\phi_j^0 \in W_h$. This formulation is equivalent to linear MsFEMs introduced earlier.

MsFEMs can be easily extended to the system of linear equations, such as elasticity equations (e.g., [235]),

$$\operatorname{div}(\mathcal{C} : E(u)) = f,$$

where $\mathcal{C}$ is the fourth-order stiffness tensor representing material properties, u is the displacement field (vector), and $E(u) = \frac{1}{2}(\nabla u + (\nabla u)^T)$ is the small strain tensor. In this case, multiscale basis functions will satisfy the local homogeneous equations $L\phi_i = 0$ in K, $\phi_i = \phi_i^0$ on ∂K, where L is the elasticity operator, $Lu = \operatorname{div}(\mathcal{C} : (\frac{1}{2}(\nabla u + (\nabla u)^T)))$. Note that the basis functions are vector fields. Vector fields ϕ_i^0 are standard finite element basis functions used for solving the system of equations. For example, for elasticity equations, ϕ_i^0 are linear functions for each element of the vector field (see [235]). The variational formulation that couples these basis functions will remain similar to (2.3) (or (2.4)).

We note that a main feature of MsFEMs presented in this book is the use of a variational formulation at the coarse scale that allows us to couple multiscale basis functions. Multiscale basis functions or multiscale maps defined by (2.12) are not necessarily based on partial differential equations and can have a discrete structure and satisfy a discrete equation at the fine grid. It is evident from the above abstract formulation that L^{map} is used only for the computation of $p_{r,h}$ (given p_h) and the variational formulation (2.13) can be chosen in different ways depending on the problem. One can consider general applications of MsFEMs involving discrete problems where the basis functions satisfy discrete systems. For example, one can consider an application where the coarse-scale equations have a continuum formulation and describe porous media flows, whereas the local problems are discrete and solved via the pore network model. MsFEMs can be used to deal with these problems.

2.5 Brief overview of various global couplings of multiscale basis functions

2.5.1 Multiscale finite volume (MsFV) and multiscale finite volume element method (MsFVEM)

Mass conservative schemes play a central role in subsurface applications. For this reason, it is important to consider methods that can provide a mass conservative approximation for the flux defined by $v = -k\nabla p$. One of these methods within a finite volume context was first proposed in [159]. The main idea of this approach is to use a finite volume global formulation with multiscale basis functions and obtain a mass conservative velocity field on a coarse grid. A similar approach was independently proposed later in [104, 133] where a finite volume element method was used. These approaches differ in their details as discussed later. In these approaches, the finite volume element method is taken as a global coupling mechanism for multiscale basis functions. The construction of basis functions remains the same as discussed earlier.

To demonstrate the concept of MsFV as well as MsFVEM, we assume $\mathcal{T}_h$ is the collection of coarse elements K. We introduce a dual grid and denote

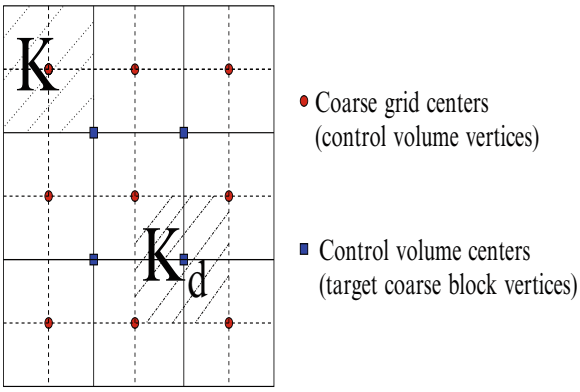

Fig. 2.6. Schematic of nodal points and coarse grids.

it by K_d (see Figure 2.6 for illustration in the case of simple rectangular grids). Furthermore, we denote the vertices of dual coarse grids by x_{K_d} (their collection by Z_{K_d}) and the vertices of target coarse-grid blocks by x_K (their collection by Z_K).

As before, the key idea of the method is the construction of basis functions on the coarse grids, such that these basis functions capture the small-scale information. As in the case of MsFEM, the basis functions are constructed from the solution of the leading-order homogeneous elliptic equation on each coarse element with some specified boundary conditions. To demonstrate MsFV, we denote by $\mathcal{P}_h$ the space spanned by the basis functions $\{\phi_j\}_{x_j \in Z_{K_d}}$ as defined before (see (2.2)). In MsFV, the basis functions on the dual grid are used and a mass conservation equation is set up on the target coarse-grid blocks. In particular, we seek $p_h \in \mathcal{P}_h$ with $p_h = \sum_{x_j \in Z^0_{K_d}} p_j \phi_j$ (where p_j are the approximate values of the solution at x_{K_d} and $Z^0_{K_d}$ is the collection of interior vertices) such that

$$\int_{\partial K} k \nabla p_h \cdot n \, dl = \int_K f dx, \tag{2.14}$$

for every target coarse-grid block $K \in \mathcal{T}_h$. Here n defines the normal vector on the boundary. The equation (2.14) results in a system of linear equations for the solution values at the nodal points of the coarse mesh. In particular, we have

$$A p_{\text{nodal}} = b,$$

where $A = (a_{ij})$, $a_{ij} = \sum_j \int_{\partial K_j} k \nabla \phi_i \cdot n dl$, $b_j = \int_{K_j} f dx$. Here j refers to the index of the coarse-grid block K_j.

In MsFVEM, the basis functions on the target coarse-grid blocks are chosen and the mass conservation equation is set up on the dual grid. We do not repeat the formulation here. The resulting multiscale method differs from the MsFEM, because it employs the finite volume or finite volume element

method as a global solver. We would like to note that the coarse-scale velocity field obtained using MsFVEM is conservative in control volume elements, whereas the velocity field obtained using MsFV is conservative in coarse elements. Further treatment is needed to obtain a conservative velocity field on the fine grid (see [159]).

Pseudo-code. We present a pseudo-code for the implementation of MsFV.

Algorithm 2.5.1

Set coarse mesh configuration from fine-scale mesh information.
For each coarse grid block n do
- For each control volume element i associated with the coarse block n
- Solve for ϕ_n^i satisfying - $L(\phi_n^i) = 0$ and boundary conditions (see (2.2))
- End for.

End do.
Assemble the mass balance equation on the coarse grid according to (2.14).
Assemble the external force on the coarse mesh according to (2.14).
Solve the coarse grid formulation.

2.5.2 Mixed multiscale finite element method

MsFV and MsFVEM introduced earlier provide a mass conservative velocity field (defined as $v = -k\nabla p$) on the coarse grid. However, the reconstructed fine-scale velocity field (using multiscale basis functions) is not conservative for the fine grid elements adjacent to coarse grid boundaries. For multiphase flow and transport simulations, the conservative fine-scale velocity is often needed. A treatment within MsFV is proposed in [159]. In this section, we present a mixed MsFEM where multiscale basis functions for the velocity field, which is highly heterogeneous, are constructed. This method allows us to achieve a mass conservative fine-scale velocity field and is used in Chapter 5 for multiphase flow simulations in heterogeneous porous media.

Our presentation of mixed MsFEM follows [71] (see also [25], [1], and [26]). First, we re-write the elliptic equation in the form

$$\begin{aligned} k^{-1}v + \nabla p = 0 \quad &\text{in} \quad \Omega \\ \operatorname{div}(v) = f \quad &\text{in} \quad \Omega \end{aligned} \tag{2.15}$$

with non-homogeneous Neumann boundary conditions $v \cdot n = g$ on $\partial\Omega$. In mixed multiscale finite element methods, the basis functions for the velocity field, $v = -k\nabla p$, are needed. As in the case of MsFEM, one can use known mixed finite element spaces to construct these basis functions. For simplicity,

we consider multiscale basis functions corresponding to lowest-order Raviart–Thomas elements (following [71], [25]). The basis functions for the velocity in each coarse block K is given by

$$\begin{aligned} \operatorname{div}(k\nabla\phi_i^K) = &\quad \frac{1}{|K|} \quad \text{in} \quad K \\ k\nabla\phi_i^K \cdot n = &\quad \begin{cases} g_i^K \text{ on } e_i^K \\ 0 \quad \text{else,} \end{cases} \end{aligned} \tag{2.16}$$

where $g_i^K = 1/|e_i^K|$ and e_i^K are the edges of K (see Figure 2.7 for the illustration). Note that these basis functions are defined for each edge by imposing constant flux along an edge (constant Neumann boundary condition) and zero flux over all other edges of the coarse-grid block. In order to preserve the total mass and have a well-posed system, some source term is needed. The source term is taken to be constant.

We define the finite-dimensional space for the velocity by

$$\mathcal{V}_h = \operatorname{span}\{\psi_i^K\},$$

where $\psi_i^K = k\nabla\phi_i^K$. For each edge e_i, one can combine the basis functions in adjacent coarse-grid blocks and obtain the basis function for the edge e_i denoted by ψ_i (or ψ_{e_i}). More precisely, if we denote by K_1 and K_2 adjacent coarse-grid blocks, then ψ_i solves (2.16) in K_1 and solves $\operatorname{div}(\psi_i) = -1/|K_2|$ in K_2, and $g_i^{K_2} = -1/|e_i|$ on $e_i^{K_2}$ and 0 otherwise. In other words, $\psi_i = \psi_i^{K_1}$ in K_1 and $\psi_i = -\psi_i^{K_2}$ in K_2, where ψ_i^K is defined via the solution of (2.16). This is illustrated in Figure 2.8. In [1], the author proposes a different construction for mixed multiscale basis functions by solving the local problem in two adjacent coarse grid blocks with zero Neumann boundary conditions and imposing positive and negative source terms. For example, $\operatorname{div}(\psi_i) = 1/|K_1|$ in K_1, $\operatorname{div}(\psi_i) = -1/|K_2|$ in K_2, and $\psi_i \cdot n = 0$ on outer boundaries of $K_1 \bigcup K_2$.

The basis functions for the pressure are piecewise constant functions over each K. We denote the span of these basis functions by Q_h. The multiscale basis functions, as in MsFEM, attempt to capture the small-scale information of the media. The functions ψ_i^K are the basis functions for the velocity field and conservative both on the fine and coarse grids provided the local problems are solved using a conservative scheme. An approximation of the fine-scale velocity field can be obtained if average fluxes along the coarse edges are known, that is if v_e is the average normal flux along the edge e and ψ_e is the corresponding basis function, then $v \approx \sum_e v_e\psi_e$ is an approximation of the fine-scale velocity field. These average fluxes, for example, can be also obtained from MsFV or MsFVEM or by using upscaling methods as in, for example, [91]. A similar idea is presented in [159] and [1]. The mixed finite element framework, presented next, couples the velocity and pressure basis functions and provides an approximation of the global solution (both p and v).

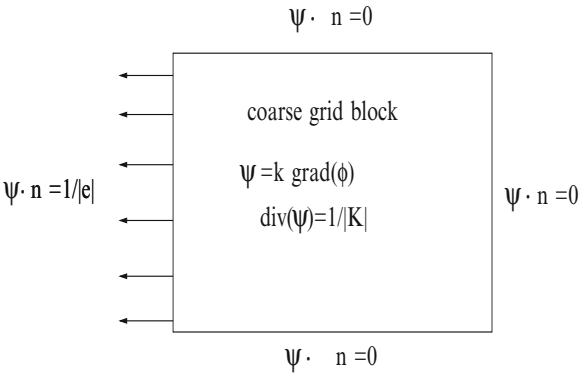

Fig. 2.7. Schematic description of a velocity basis function construction in a coarse grid block.

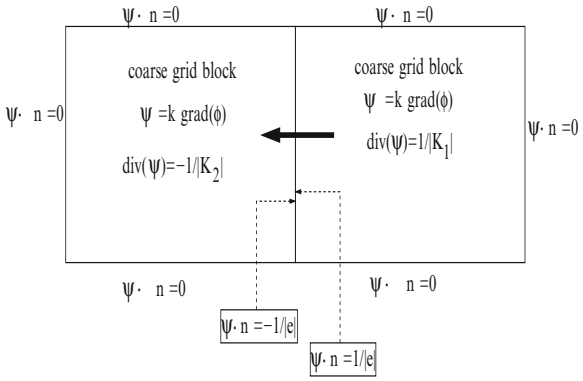

Fig. 2.8. Schematic description of a velocity basis function for an edge combining adjacent basis functions.

To formulate the mixed MsFEM, we use the numerical approximation associated with the lowest-order Raviart–Thomas mixed finite element to find $\{v_h, p_h\} \in \mathcal{V}_h \times Q_h$ such that $v_h \cdot n = g_h$ on $\partial\Omega$, where $g_h = g_{0,h} \cdot n$ on $\partial\Omega$ and $g_{0,h} = \sum_{e \in \{\partial K \bigcap \partial\Omega, K \in \mathcal{T}_h\}} (\int_e g ds)\psi_e$, $\psi_e \in \mathcal{V}_h$ is the corresponding basis function to edge e,

$$\begin{aligned} \int_\Omega k^{-1} v_h \cdot w_h dx - \int_\Omega \mathrm{div}(w_h) p_h dx = 0, \quad \forall w_h \in \mathcal{V}_h^0 \\ \int_\Omega \mathrm{div}(v_h) q_h dx = \int_\Omega f q_h dx, \quad \forall q_h \in Q_h, \end{aligned} \tag{2.17}$$

where $\mathcal{V}_h^0$ is a subspace of $\mathcal{V}_h$ with elements that satisfy homogeneous Neumann boundary conditions. The above formulation was the mixed MsFEM introduced in [71].

The discrete formulation of (2.17) can be easily written down as

$$\begin{bmatrix} A & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} v^D \\ -p^D \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}, \tag{2.18}$$

where $v = \sum_i v_i^D \psi_i$ and $p = \sum_i p_i^D q_i$ with v_i^D being normal interface fluxes and p_i^D being the cell average solution. Here $A = (a_{ij})$, $C = (c_{ik})$, and $b = (b_k)$ are defined by

$$a_{ij} = \int_\Omega \psi_i \cdot (k)^{-1} \psi_j dx, \quad c_{ik} = \int_\Omega q_k \operatorname{div}(\psi_i) dx \text{ and } b_k = \int_\Omega q_k f dx.$$

This linear system is indefinite, and thus it is in general harder to solve than the positive definite systems that arise (e.g., from Galerkin finite element discretizations). However, it is common to solve the mixed linear system (2.18) by using a so-called hybrid formulation. In the hybrid formulation the system (2.18) is localized by introducing an extra set of unknowns representing p at the grid cell interfaces. By performing some simple algebraic manipulations, we then obtain a positive definite system that is solved for the interface pressures. Finally, the solution to (2.18) is computed from the solution to the hybrid system by performing local algebraic calculations.

We note that in [10], a discontinuous Galerkin method has been used as a global coupling for multiscale basis functions and discontinuous Galerkin MsFEM is proposed. We refer to [10] for details. We also refer to [16], for the use of discontinuous Galerkin method within the framework of HMM.

Pseudo-code. Next, we briefly outline the implementation of mixed MsFEM. We have put simple prototype MATLAB codes for solving elliptic equations with mixed MsFEM (courtesy of J. Aarnes) at
http : *//www.math.tamu.edu/* $\sim$ *yalchin.efendiev/codes/*.

Algorithm 2.5.2

Set coarse mesh configuration from fine-scale mesh information.
For each coarse-grid block n do
- For each edge of a coarse-grid block
- Solve for ψ_n^i according to (2.16)
- End for

End do.
Assemble the coarse-scale system according to (2.17).
Assemble the external force on the coarse mesh according to (2.17).
Solve the coarse grid formulation.

Comments on the assembly of stiffness matrix. Similar to the Galerkin MsFEM, one can use the representation of multiscale basis functions via fine-scale basis functions to assemble the matrices in (2.18). It can be shown that

$A = DA^f D^T$, where $D = (d_{ij})$ is a matrix defined by

$$\psi_i = d_{ij}\psi_j^{0,f}$$

with $\psi_j^{0,f}$ being fine-scale basis functions, $A^f = (a_{lm}^f)$, $a_{lm}^f = \int_\Omega \psi_l^{0,f} \cdot (k)^{-1}\psi_m^{0,f} dx$. This simplification can be used in the assembly of the stiffness matrix.

2.6 MsFEM for problems with scale separation

In some applications, regions smaller than the coarse-grid block are sufficient to represent the small-scale effects. In these applications, one can use the basis functions constructed in a smaller region, instead of the coarse-grid block, to capture the small-scale effects. The basic idea behind this localization is that $(1/|K|)\int_K k\nabla\phi_i dx$ in the computation of the stiffness matrix (2.4) can be approximated by $(1/|K_{\text{loc}}|)\int_{K_{\text{loc}}} k\nabla\tilde{\phi}_i dx$, where ϕ_i is the solution of $\text{div}(k\nabla\phi_i) = 0$ in K, $\phi_i = \phi_i^0$ on ∂K, and $\tilde{\phi}_i$ is the solution of $\text{div}(k\nabla\tilde{\phi}_i) = 0$ in K_{loc}, $\tilde{\phi}_i = \phi_i^0$ on ∂K_{loc}. Here, K_{loc} refers to a smaller region (RVE) as before. Next, we briefly discuss this approximation. Within the general G-convergence theory (e.g., [164]), it can be shown that (e.g., [164])

$$\lim_{\epsilon\to 0}\frac{1}{|K|}\int_K k\nabla\phi_i dx = \frac{1}{|K|}\int_K k^*\nabla\phi_i^0 dx, \tag{2.19}$$

where ϵ is the characteristic length scale and ϕ_i^0 is the homogenized part of the basis function. Note that the G-convergence theory does not assume periodicity and $k^*(x)$ are homogenized coefficients independent of ϵ such that the solution and the fluxes of the homogenized equation $-\text{div}(k^*(x)\nabla p^*) = f$ approximate the solution of fine-scale equation $-\text{div}(k(x)\nabla p) = f$ in appropriate norms (we refer to, e.g., [164] for details). The same result holds for $(1/|K_{\text{loc}}|)\int_{K_{\text{loc}}} k\nabla\tilde{\phi}_i dx$; that is

$$\lim_{\epsilon\to 0}\frac{1}{|K_{\text{loc}}|}\int_{K_{\text{loc}}} k\nabla\tilde{\phi}_i dx = \frac{1}{|K_{\text{loc}}|}\int_{K_{\text{loc}}} k^*\nabla\tilde{\phi}_i^0 dx. \tag{2.20}$$

Assuming that $k^*(x)$ is sufficiently smooth, one can approximate $k^*(x)$ within each coarse block by a constant and show that the right-hand sides of (2.19) and (2.20) are close for small h (note that $K_{\text{loc}} \subset K$). Consequently, for small ϵ, the left-hand sides of (2.19) and (2.20) will be close; that is

$$\lim_{|K|\to 0}\lim_{\epsilon\to 0}\frac{1}{|K|}\int_K k\nabla\phi_i dx = \lim_{|K_{\text{loc}}|\to 0}\lim_{\epsilon\to 0}\frac{1}{|K_{\text{loc}}|}\int_{K_{\text{loc}}} k\nabla\tilde{\phi}_i dx. \tag{2.21}$$

The relation (2.21) shows that $\int_{K_{\text{loc}}} k\nabla\tilde{\phi}_i dx$ can be used to approximate $\int_K k\nabla\phi_i dx$ in the limit $\lim_{h\to 0}\lim_{\epsilon\to 0}$ (limit of scale separation). From here,

one can show that the multiscale finite element solution approximates the fine-scale solution in $\lim_{h\to 0}\lim_{\epsilon\to 0}$. We refer to [108] for the details where the more general problem is studied. For periodic problems, this approximation can be shown in the limit $\epsilon/h_{\text{loc}} \to 0$ (where h_{loc} is the size of K_{loc}) if K_{loc} is much larger than the period size. If $k^*(x)$ is sufficiently smooth and ϕ_i^0 is piecewise linear, then (2.21) is equal to $k^*(x_0)\nabla\phi_i^0$, where x_0 is the point to which the region K or K_{loc} contracts (see [164] for a more precise definition of x_0). In this case, the location of RVE within the coarse grid block is not very important and one can choose RVE, for example, at the center of the mass of the coarse element.

In the case of periodic heterogeneities, where the period is known, the basis functions can be approximated using homogenization theory by

$$\phi_j \approx \phi_j^0 + \epsilon\chi^i \frac{\partial}{\partial x_i}\phi_j^0, \tag{2.22}$$

where the summation over repeated indices occurs. This approach derives from homogenization and χ is a periodic solution (with average zero) of (2.9). Consequently, the approximation of the basis functions can be carried out in a domain of the size of the period that characterizes the small-scale oscillation of $k(x)$. This reduces the computational cost if the period is much smaller than the coarse-grid block. With this approximation, the stiffness matrix (see (2.5)) can be assembled in the periods instead of the coarse grid blocks:

$$a_{ij} = \sum_K \int_K k\nabla\phi_i \cdot \nabla\phi_j^0 dx \approx \sum_K \frac{|K|}{|Y|}\int_Y k\nabla(\phi_j^0 + \epsilon\chi^i\frac{\partial}{\partial x_i}\phi_j^0)\cdot\nabla\phi_j^0 dx, \tag{2.23}$$

where Y is the period within K. One can further approximate this expression and show that $a_{ij} \approx \sum_K (|K|/|Y|)\int_Y k(I+\nabla\chi)\nabla\phi_j^0\cdot\nabla\phi_j^0 dx$. In [269], the author uses the approximation of the basis functions based on (2.22) for periodic coefficients and shows that MsFEM converges as the period size and the mesh size go to zero.

As we discussed earlier, using multiscale basis functions, a fine-scale approximation of the solution can be easily computed, by $p_h = \sum_i p_i\phi_i$. When regions smaller than the coarse-grid block are used for computing basis functions, then the basis functions can be extended to a coarse-grid block based on homogenization expansion. In particular, from the problem in K_{loc} (RVE) one can easily extract χ and use it to construct an extension of the basis function to the coarse-grid block. These basis functions can be further used to obtain an approximate fine-scale solution in the entire domain.

We would like to note that this approximation procedure is not limited to periodic problems and can be applied to problems where homogenization by periodization (see the principle of periodic localization [164]) is true. The random homogeneous case with ergodicity is one of these cases. The techniques discussed in this section can be also used in MsFV, MsFEM, mixed MsFEM,

and other multiscale methods when the problem has some special features such as periodicity or strong scale separation.

2.7 Extension of MsFEM to parabolic problems

MsFEM can be naturally extended to parabolic equations. In this section, we briefly describe the extension of MsFEM to the parabolic equation

$$\frac{\partial}{\partial t}p(x,t) - \operatorname{div}(k(x,t)\nabla p(x,t)) = f \tag{2.24}$$

with appropriate boundary conditions on the finite time interval $[0,T]$ and smooth initial conditions. In general, when there are space and time heterogeneities, basis functions are the solutions of the leading-order homogeneous parabolic equations. In the absence of time heterogeneities (i.e., $k(x,t) = k(x)$), one can use spatial basis functions developed for elliptic equations. To introduce MsFEM, we assume for simplicity that the interval $[0,T]$ is divided into M equal parts $0 = t_0 < t_1 < \cdots < t_M = T$. These intervals are coarse-scale intervals; that is $\Delta t = t_{i+1} - t_i$ is larger than the characteristic time scale. The basis functions are constructed in $[t_n, t_{n+1}]$ as the solution of

$$\frac{\partial}{\partial t}\phi_i(x,t) - \operatorname{div}(k(x,t)\nabla\phi_i(x,t)) = 0 \tag{2.25}$$

in each K such that $\phi_i = \phi_i^0$ on ∂K and $\phi_i(x, t = t_n) = \phi_i^0$. Here, $\phi_i^0 \in W_h$ are standard finite element basis functions (e.g., piecewise linear functions). We seek the finite-dimensional approximation of the solution in $[t_n, t_{n+1}]$ as

$$p_h^{n+1}(x,t) = \sum_i p_i^{n+1}\phi_i(x,t), \tag{2.26}$$

where p_i^{n+1} (approximate nodal values of the solution) will be determined. Then, substituting (2.26) into the original equation, multiplying it by ϕ_i^0 (as in the Petrov–Galerkin formulation) and integrating over the space and $[t_n, t_{n+1}]$, we have

$$\begin{aligned} &p_i^{n+1}\int_\Omega \phi_i(x,t_{n+1})\phi_j^0(x)dx - p_i^n\int_\Omega \phi_i(x,t_n)\phi_j^0(x)dx \\ &+ \int_{t_n}^{t_{n+1}}\sum_K \int_K k(x,t)\nabla p_h(x,t)\cdot\nabla\phi_j^0(x)dxdt = \int_{t_n}^{t_{n+1}}\int_\Omega f\phi_j^0(x)dxdt, \end{aligned} \tag{2.27}$$

where $p_h(x,t) = \sum_i p_i(t)\phi_i(x,t)$. The third term on the right-hand side can be treated implicitly or explicitly. In particular, the implicit method is given by

$$p_i^{n+1}\int_\Omega \phi_i(x,t_{n+1})\phi_j^0(x)dx - p_i^n\int_\Omega \phi_i(x,t_n)\phi_j^0(x)dx$$
$$+\, p_i^{n+1}\int_{t_n}^{t_{n+1}}\sum_K\int_K k(x,t)\nabla\phi_i(x,t)\cdot\nabla\phi_j^0(x)dxdt = \int_{t_n}^{t_{n+1}}\int_\Omega f\phi_j^0(x)dxdt. \tag{2.28}$$

If the third term is evaluated explicitly, that is it is replaced by

$$p_i^n\int_{t_n}^{t_{n+1}}\sum_K\int_K k(x,t)\nabla\phi_i(x,t)\cdot\nabla\phi_j^0(x)dxdt,$$

then the resulting method is explicit. One can easily write down the corresponding discrete formulation which we omit here.

We note that if there are no temporal heterogeneities (i.e., $k(x,t)=k(x)$), the basis functions can be the solutions of elliptic equations as in the case of elliptic equations. The equations for the basis functions can be simplified depending on the relation between spatial and temporal heterogeneities (see Section 3.5). Equation (2.25) defines the basis functions independent of the relation between spatial and temporal heterogeneities. Furthermore, in the case of scale separation, (2.25) can be solved in a smaller volume, RVE, and this solution can be used in Equation (2.27) in a manner similar to the elliptic case.

Finally, we would like to note that one can couple the basis functions using different methods, such as finite volume element methods and so on. For example, the mixed MsFEM for parabolic equations (with time-independent coefficients, $k(x,t)=k(x)$) has the following formulation. We seek $\{v_h,p_h\}\in\mathcal{V}_h\times Q_h$, such that

$$\begin{aligned}\int_\Omega \frac{\partial p_h}{\partial t}q_h dx + \int_\Omega \mathrm{div}(v_h)\; q_h dx &= \int_\Omega f q_h dx, \quad \forall q_h\in Q_h\\ \int_\Omega k^{-1}v_h\cdot w_h dx - \int_\Omega \mathrm{div}(w_h)\; p_h dx &= 0, \quad \forall w_h\in\mathcal{V}_h^0,\end{aligned} \tag{2.29}$$

where $\mathcal{V}_h$, $\mathcal{V}_h^0$, and Q_h are defined as before (cf. (2.17)) for elliptic equations.

2.8 Comparison to other multiscale methods

MsFEMs share similarities with many other multiscale methods. One of the early approaches is the upscaling technique (e.g., [91, 260, 47]) which is based on the homogenization method. The main idea of upscaling techniques is to form a coarse-scale equation and pre-compute the effective coefficients. In the case of linear elliptic equations, the coarse-scale equation has the same form as the fine-scale equation except that the coefficients are replaced by effective homogenized coefficients. The effective coefficients in upscaling methods are

computed using the solution of the local problem in a representative volume. Various boundary conditions can be used for solving the local problems and, for simplicity, we consider

$$\operatorname{div}(k\nabla\phi_e) = 0 \text{ in } K \tag{2.30}$$

with $\phi_e(x) = x\cdot e$ on ∂K, where e is a unit vector. It is sufficient to solve (2.30) for d linearly independent vectors $e_1, ..., e_d$ in $\mathbb{R}^d$ because $\phi_e = \sum_i \beta_i \phi_{e_i}$ if $e = \sum_i \beta_i e_i$. Here K denotes a coarse-grid block, although one can use a smaller region as discussed in Section 2.6. The effective coefficients are computed in each K as

$$\tilde{k^*}e = \frac{1}{|K|}\int_K k\nabla\phi_e dx. \tag{2.31}$$

We note that $\tilde{k^*}$ (which is not the same as the homogenized coefficients) is a symmetric matrix provided k is symmetric and (2.31) can be computed for any point in the domain by placing the point at the center of K, i.e.,

$$\tilde{k^*}(x_0)e = \frac{1}{|K_{x_0}|}\int_{K_{x_0}} k\nabla\phi_e dx,$$

where K_{x_0} is the RVE with the center at x_0 and ϕ_e is the local solution defined by (2.30) in K_{x_0}. One can use various boundary conditions, including periodic boundary conditions as well as oversampling methods. We refer to [91, 260] for the discussion on the use of various boundary conditions. Once the effective coefficients are calculated, the coarse-scale equation

$$-\operatorname{div}(\tilde{k^*}\nabla p^*) = f \tag{2.32}$$

is solved over the entire region.

To show the similarity to MsFEMs, we write down the discretization of (2.32) using the Galerkin finite element method. Find $p_h^* \in W_h$, such that

$$\sum_K \int_K \tilde{k^*}\nabla p_h^* \cdot \nabla v_h dx = \int_\Omega f v_h dx, \quad \forall v_h \in W_h. \tag{2.33}$$

Next, we write down the Petrov–Galerkin MsFEM discretization (see (2.4))

$$a_{ij}p_i = b_j, \tag{2.34}$$

where $a_{ij} = \sum_K \int_K k\nabla\phi_i dx \cdot \nabla\phi_j^0$ (assuming ϕ_j^0 is piecewise linear) and $b_j = \int_\Omega f\phi_j^0 dx$. One can show that

$$a_{ij} = \sum_K \int_K \tilde{k^*}\nabla\phi_i^0 \cdot \nabla\phi_j^0 dx$$

because $(1/|K|)\int_K k\nabla\phi_i dx = \tilde{k^*}\nabla\phi_i^0$. We assumed that ϕ_i^0 are piecewise linear functions. Thus, (2.34) and (2.33) are equivalent. This shows that the

MsFEM can be derived from traditional upscaling methods. However, the concept of MsFEMs differs from traditional upscaling methods, because the local information is directly coupled via a variational formulation and we do not assume a specific form for coarse-scale equations. Moreover, MsFEMs allow us to recover the local information adaptively which makes it a powerful tool (e.g., for porous media flow simulations). More advantages of MsFEM are discussed in later chapters.

Next, we briefly discuss the relation between variational multiscale approaches and MsFEM. These similarities are also shown in [26] in the context of mixed finite element methods. Here, we discuss Galerkin finite element methods. We assume that the fine-scale solution space X_F is partitioned into the coarse-dimensional space X_C (e.g., W_h), and the space containing the unresolved scales X_U,

$$X_F = X_C \oplus X_U.$$

We assume also that these spaces are the subspaces of $H_0^1(\Omega)$, for simplicity. The fine-scale solution can be written accordingly as

$$p = p_C + p_U.$$

Substituting this into the original equation and multiplying by the test functions from X_C, we obtain the equation for the coarse-scale solution

$$\int_\Omega k\nabla(p_C + p_U) \cdot \nabla v_h dx = \int_\Omega f v_h dx, \quad \forall v_h \in X_C. \tag{2.35}$$

Similarly, multiplying the original equation by the test functions from X_U, we obtain the equation for the unresolved part of the solution

$$\int_\Omega k\nabla p_U \cdot \nabla v_h dx = \int_\Omega f v_h dx - \int_\Omega k\nabla p_C \cdot \nabla v_h dx, \quad \forall v_h \in X_U. \tag{2.36}$$

To find the coarse-scale solution, p_C, one first solves p_U from (2.36) and substitutes it into (2.35) to compute p_C. We note that (2.35) is exact and the solution of (2.36) is nonlocal. In general, the solution of (2.36) can be localized by imposing local boundary conditions. One can use various choices for the boundary conditions. Noting the solution of the local problem can be written via generic basis functions, one can derive a formulation similar to MsFEM.

To show the similarity between MsFEMs and variational multiscale methods, as an example, we consider the localization based on $p_U = 0$ on the boundaries of the coarse-grid block K. In this case, it is evident that the solution $p_C + p_U$ satisfies the local problem $\mathrm{div}(k\nabla(p_C + p_U)) = f$ in K and $p_C + p_U$ is a piecewise linear function on ∂K. This solution can be approximated by multiscale finite element basis functions defined by (2.2). Thus, we can seek $p_C + p_U = \sum_i p_i \phi_i$. Substituting this expression into (2.35), we obtain a Petrov–Galerkin formulation of MsFEM if ϕ_i are chosen with zero right-hand side. We note that one of the differences between the variational

multiscale method and MsFEM is that the former uses source terms in the formulation of the local problems. The representation of source terms with MsFEMs in the context of subsurface flows has been extensively studied in the literature (e.g., see [13, 175], Sections 5.5, and 5.6) within the context of subsurface flows. Typically, only singular source terms require special treatment with multiscale basis functions.

As we mentioned earlier, one can take advantage of scale separation in MsFEM. There are various ways to do so and these approaches will share similarities, for example, with the application of heterogeneous multiscale methods (HMM) ([97]), and multiscale enrichment methods ([121]). HMM has been extensively studied in the literature (e.g., see [17, 15, 98, 203] for the applications to elliptic equations). The main idea of this approach is to use small regions at quadrature points for the computation of effective coefficients. This is performed on-the-fly when the stiffness matrix corresponding to the coarse-scale problem is assembled. As mentioned above, multiscale basis functions can be approximated when there is scale separation. The basic idea behind this localization is that $(1/|K|)\int_K k\nabla\phi_i dx$ (in the stiffness matrix, see (2.5)) can be approximated by $(1/|K_{\text{loc}}|)\int_{K_{\text{loc}}} k\nabla\tilde{\phi}_i dx$, where ϕ_i is the solution of $\text{div}(k\nabla\phi_i) = 0$ in K, $\phi_i = \phi_i^0$ on ∂K, and $\tilde{\phi}_i$ is the solution of $\text{div}(k\nabla\tilde{\phi}_i) = 0$ in K_{loc}, $\tilde{\phi}_i = \phi_i^0$ on ∂K_{loc}. Using the general G-convergence theory (e.g., [164]), one can show (2.21). This result holds when $k^*(x)$ is a smooth function and (2.21) is equal to $k^*(x_0)\nabla\phi_i^0$ (assuming ϕ_i^0 is piecewise linear) at almost every point x_0 to which the region K and K_{loc} contract. For periodic problems, $k = k(x/\epsilon)$, it is not difficult to show that

$$|\frac{1}{|K|}\int_K k\nabla\phi_i dx - k^*\nabla\phi_i^0| \leq C(\frac{\epsilon}{h} + h),$$

where k^* is computed for the coarse-grid block according to (2.31). Similarly,

$$|\frac{1}{|K_{\text{loc}}|}\int_{K_{\text{loc}}} k\nabla\tilde{\phi}_i dx - k^*\nabla\phi_i^0| \leq C(\frac{\epsilon}{h_{\text{loc}}} + h_{\text{loc}}).$$

Based on these results, one can show the convergence of MsFEMs using the local information in K_{loc}. We refer to [108] for the details where a more general problem is studied. This approximation of the basis functions and the corresponding approximation of the stiffness matrix elements can save CPU time if there is a strong scale separation. The method obtained in this way is very similar to the application of HMM to elliptic equations, although it differs in some details (e.g., the computations are not performed at quadrature points). We would like to note that one can also use first-order corrector approximation for the basis functions as discussed earlier. In this case, the local solution in RVE can be used as a cell problem solution $\epsilon\chi$ in (2.22). We would like to mention that there are other approaches (e.g., [121, 122, 153]) which use the solution of the cell problem to construct multiscale basis functions based on the partition of unity method.

As we mentioned in Section 1.2, multiscale methods considered in this book differ from domain decomposition methods (e.g., [257]) where the local problems are solved many times iteratively to obtain an accurate approximation of a fine-scale solution. Multiscale methods studied in this book share similarities with upscaling/homogenization methods, where the basis functions are computed based on coarse-grid information. Figure 2.9 illustrates the main concept of the MsFEM and its advantages (see also Section 2.9). The multiscale methods attempt to find the coarse-scale solution and can also compute an approximation of the fine-scale solution via downscaling. One can use iterations (e.g., [93]) similar to domain decomposition methods or some type of global information to improve the accuracy of multiscale methods when there is no scale separation (see Chapter 4).

Finally, we remark that we restricted ourselves only to a few multiscale methods due to the page limitation. One can find similarities between multiscale finite element methods and other multiscale methods known in the literature. Some of these similarities may not be so apparent. Some of these algorithms are designed for periodic problems and have advantages when the underlying heterogeneities are periodic. For example, the approach proposed in [198] is based on two-scale convergence concept ([21]). This approach is generalized to problems with multiple separable scales ([139]) using hierarchical basis functions. In this book, we do not want to discuss the similarities between different multiscale methods to a great extent and instead focus on our work on extensions and applications of various MsFEMs. We again stress that the main idea of MsFEM stems from the earlier work of Babuška and Osborn [33]. In Chapters 3 and 4, we show that the MsFEM can take advantage of global information and can be naturally extended to nonlinear problems.

2.9 Performance and implementation issues

We outline the implementation of a Galerkin MsFEM for a simple test problem (following [145]) and define some notations that are used in the discussion below. We consider solving problems in a unit square domain. Let N be the number of elements in each coordinate direction. The mesh size is thus $h = 1/N$. To compute the basis functions, each element is discretized into $M \times M$ subcell elements with mesh size $h_f = h/M$. To implement the oversampling method, we partition the domain into sampling domains where each of them contains many elements. Analysis and numerical tests indicate that the size of the sampling domains can be chosen freely as long as the boundary layer is avoided. In practice, though, one wants to maximize the efficiency of oversampling by choosing the largest possible sample size that reduces the redundant computation of overlapping domains to a minimum.

In general, the multiscale basis functions are constructed numerically, except for certain special cases. They are solved in each K or K_E using a standard FEM. The global linear system on Ω is solved using the same method.

Numerical tests show that the accuracy of the final solution is weakly insensitive to the accuracy of basis functions.

Because the basis functions are independent of each other, their construction can be carried out in parallel perfectly. In a parallel implementation of oversampling, the sample domains are chosen such that they can be handled within each processor without communication. More implementation details can be found in [145].

2.9.1 Cost and performance

In computations, a large amount of overhead time comes from constructing the basis functions. This is also true for classical upscaling methods discussed in Section 2.8. On a sequential machine, the operation count of the MsFEM is about twice that of a conventional FEM for a 2D problem. However, due to good parallel efficiency, this difference is reduced significantly on a massively parallel computer (see [145] for a detailed study of the MsFEM's parallel efficiency). This overhead can be reduced if there is scale separation.

In practice, multiple solves are often required for different source terms, boundary conditions, mobilities and so on. MsFEMs have advantages in such situations and the overhead of basis construction can be negligible because the basis functions can be re-used. This is illustrated in Figure 2.9, where pre-computed multiscale basis functions can be re-used for different external parameters such as source terms, boundary conditions and the like. Moreover, multiscale basis functions can be used to re-construct the fine-scale features of the solution in the regions of interest. This adaptivity is often used in subsurface applications where the fine-scale features of the velocity $(-k\nabla p)$ are re-constructed in some regions where the detailed velocity information is needed, for example, for updating sharp interfaces. In summary, MsFEMs provide the following advantages in simulations: (1) parallel multiscale basis function construction (which can be very cheap if there is scale separation); (2) re-use of basis functions for different external parameters and inexpensive coarse-scale solve; and (3) adaptive downscaling of the fine-scale features of the solution in the regions of interest.

Significant computational savings are obtained for time-dependent problems such as those that occur in subsurface applications. In these problems, the heterogeneities representing porous media properties do not change and the basis functions are pre-computed at the initial time. These basis functions are used throughout the simulations and the elliptic equations are solved on the coarse grid each time. In this sense, our approaches are similar to classical upscaling methods where the upscaled quantities are pre-computed before solving the equations on the coarse grid. In some situations, local basis function update is required, for example, if there is a sharp interface dividing two propagating fluids. The interface modifies the permeability and this requires local updates of the basis functions.

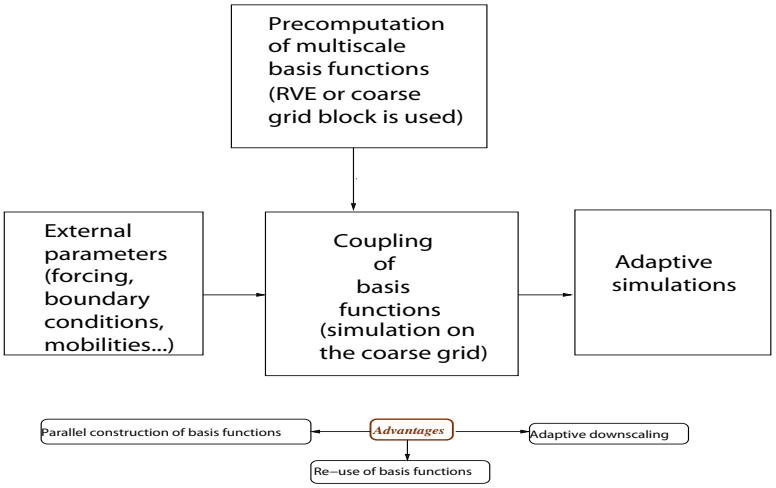

Fig. 2.9. A schematic illustration of multiscale simulations and advantages.

2.9.2 Convergence and accuracy

Because we need to use an additional grid to compute the basis function numerically, it makes sense to compare our MsFEM with a traditional FEM at the subcell (fine) grid, $h_f = h/M$. Note that the MsFEM captures the solution at the coarse grid h, whereas FEM tries to resolve the solution at the fine grid h_f. Our extensive numerical experiments demonstrate that the accuracy of the MsFEM on the coarse grid h is comparable to that of FEM on the fine grid.

As an example, in Table 2.9.2 we present the results from [145] for

$$k(x/\epsilon) = \frac{2 + A\sin(2\pi x_1/\epsilon)}{2 + A\cos(2\pi x_2/\epsilon)} + \frac{2 + \sin(2\pi x_2/\epsilon)}{2 + A\sin(2\pi x_1/\epsilon)} \quad (A = 1.8), \tag{2.37}$$

$$f(x) = -1 \qquad \text{and} \qquad p|_{\partial\Omega} = 0. \tag{2.38}$$

The convergence of three different methods is compared for fixed $\epsilon/h = 0.64$, where "-L" indicates that a linear boundary condition is imposed on the multiscale basis functions, "os" indicates the use of oversampling, and LFEM stands for the standard FEM with bilinear basis functions.

We see clearly the scale resonance in the results of MsFEM-L and the (almost) first-order convergence (i.e., no resonance) in MsFEM-os-L. The error of MsFEM-os-L is smaller than that of LFEM obtained on the fine grid. In [147, 145], more extensive convergence tests have been presented.

There have been many numerical studies of MsFEM, in particular, in the context of multiphase flow simulations. Some of these results are presented and discussed in the book.

Table 2.1. Convergence for periodic case

N	ϵ	MsFEM-L		MsFEM-os-L		LFEM	
		$\|\|E\|\|_{l^2}$	Rate	$\|\|E\|\|_{l^2}$	Rate	MN	$\|\|E\|\|_{l^2}$
16	0.04	3.54e-4		7.78e-5		256	1.34e-4
32	0.02	3.90e-4	-0.14	3.83e-5	1.02	512	1.34e-4
64	0.01	4.04e-4	-0.05	1.97e-5	0.96	1024	1.34e-4
128	0.005	4.10e-4	-0.02	1.03e-5	0.94	2048	1.34e-4

2.9.3 Coarse-grid choice

We would like to remark that in MsFEM simulations, one is not restricted to rectangular or box-shaped coarse and fine grids. In fact, as demonstrated in a number of papers [11, 5], one can use an unstructured fine grid. Moreover, the coarse grid can have an arbitrary shape and the only requirement on the coarse grid is that every coarse grid consists of a connected union of fine-grid blocks. In Figure 2.10, we present an example from [11]. As one can observe from this figure the coarse-grid blocks have quite irregular shapes. In [9], the authors develop gridding techniques that use single-phase flow information (surrogate global information) to construct a coarse grid. The coarse grid is chosen such that it minimizes the magnitude of the single-phase velocity field variation within each coarse-grid block. This automatic coarse-grid generator allows one to use an optimal coarse grid for accurate simulation purposes in two-phase flow simulations. In Chapter 4, we discuss an extension of mixed MsFEM to unstructured coarse grids and include a few numerical examples to demonstrate its effectiveness. We would like to note that the fine grid blocks in neighboring coarse-grid blocks do not need to match along the interface.

In general, an appropriate choice of the coarse-grid will improve the efficiency and accuracy of multiscale methods. It is often possible that the solution may have smooth variation along coarse grid boundaries if the coarse grid is judiciously selected. This can lead to improved numerical results. Some of these issues are discussed in Chapter 4. For computational purposes, it is important that the coarse grid is more regular (for accuracy purposes) and the number of fine-grid blocks within a coarse grid is approximately the same (for load-balancing purposes).

2.10 An application to two-phase flow

MsFEMs and their modifications have been used with success in two-phase flow simulations through heterogeneous porous media. First, we briefly describe the underlying fine-scale equations. We present two-phase flow equations neglecting the effects of gravity, compressibility, capillary pressure, and dispersion on the fine scale. Porosity, defined as the volume fraction of the void

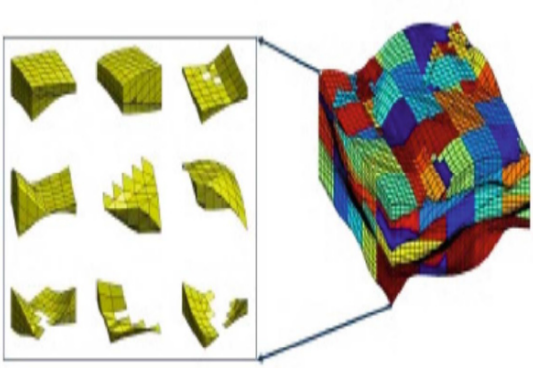

Fig. 2.10. Unstructured fine and coarse grids (from [11]).

space, is taken to be constant and therefore serves only to rescale time. The two phases are referred to as water and oil and designated by the subscripts w and o, respectively. We can then write Darcy's law, with all quantities dimensionless, for each phase j $(j = w, o)$ as follows;

$$v_j = -\lambda_j(S)k\nabla p, \tag{2.39}$$

where v_j is phase velocity, S is water saturation (volume fraction), p is pressure, $\lambda_j = k_{rj}(S)/\mu_j$ is phase mobility, where k_{rj} and μ_j are the relative permeability and viscosity of phase j, respectively, and k is the permeability tensor.

Combining Darcy's law with conservation of mass, $\mathrm{div}(v_w + v_o) = 0$, allows us to write the flow equation in the following form

$$\mathrm{div}(\lambda(S)k\nabla p) = q_t, \tag{2.40}$$

where the total mobility $\lambda(S)$ is given by $\lambda(S) = \lambda_w(S) + \lambda_o(S)$ and q_t is a source term representing wells/sources. The term $q_t = q_w + q_o$ represents the total volumetric source term. The saturation dynamics affects the flow equations. One can derive the equation describing the dynamics of the saturation

$$\frac{\partial S}{\partial t} + \mathrm{div}(v f_w(S)) = -q_w, \tag{2.41}$$

where $f_w(S)$ is the fractional flow of water, given by $f_w = \lambda_w/(\lambda_w + \lambda_o)$. The signs of the source terms that appear in (2.40) and (2.41) can be inter-changed. The total velocity v is given by

$$v = v_w + v_o = -\lambda(S)k\nabla p. \tag{2.42}$$

In the presence of capillary effects, an additional degenerate diffusion term is present in (2.41).

If $k_{rw} = S$, $k_{ro} = 1 - S$, and $\mu_w = \mu_o$, then the flow equation reduces to

$$\operatorname{div}(k\nabla p^{sp}) = q_t.$$

This equation is called single-phase flow equation.

For two-phase flow simulations, we first solve the coarse-scale pressure equation using MsFVEM. More precisely, assuming that the solution $S(x,t)$ is known at time $t = t_n$, we solve $\operatorname{div}(\lambda(S(x,t_n))k\nabla p) = q_t$ using MsFVEM to compute $p(x,t_{n+1})$ on the coarse grid. The fine-scale velocity $v(x,t_{n+1})$ is then re-constructed by solving a local fine-scale problem over each dual cell with flux boundary conditions, as determined from the pressure solution. This velocity is then used in the explicit solution of the saturation equation using a first-order upwind method to compute $S(x,t_{n+1})$. The overall procedure is thus an IMPES (implicit pressure, explicit saturation) approach. We also consider an approach where the coarse-scale velocity is used to update the saturation field.

As we see from (2.40) and (2.41), the pressure equation is solved many times for different saturation profiles. Thus, computing the basis functions once at time zero is very beneficial and the basis functions are only updated near sharp interfaces. In fact, our numerical results show that only slight improvement can be achieved by updating the basis functions near sharp fronts.

We present a representative numerical example for a permeability field generated using two-point geostatistics. To generate this permeability field, we have used the GSLIB algorithm [85]. The permeability is log-normally distributed with prescribed variance $\sigma^2 = 1.5$ (σ^2 here refers to the variance of $\log k$) and some correlation structure. The correlation structure is specified in terms of dimensionless correlation lengths in the x_1- and x_2-directions, $l_1 = 0.4$, and $l_2 = 0.04$, nondimensionalized by the system length. Linear boundary conditions are used for constructing multiscale basis functions. A spherical variogram is used. In this numerical example, the fine-scale field is 120×120, and the coarse-scale field is 12×12 defined in the rectangle with the length 5 and the width 1. For the two-phase flow simulations, the system is considered to initially contain only oil ($S = 0$) and water is injected at inflow boundaries ($S = 1$ is prescribed); that is we specify $p = 1$, $S = 1$ along the $x = 0$ edge and $p = 0$ along the $x = 5$ edge, and no flow boundary conditions on the lateral boundaries. Relative permeability functions are specified as $k_{rw} = S^2$, $k_{ro} = (1 - S)^2$; water and oil viscosities are set to $\mu_w = 1$ and $\mu_o = 5$. Source terms q_w and q_t are zero. Results are presented in terms of the fraction of oil in the produced fluid, called fractional flow or oil-cut (designated F), against pore volume injected (PVI). Fractional flow is given by

$$F(t) = 1 - \frac{\int_{\partial\Omega^{\text{out}}} (v \cdot n) f(S) ds}{\int_{\partial\Omega^{\text{out}}} v \cdot n ds}, \tag{2.43}$$

where Ω^{out} refers to the part of the boundary with outer flow; that is $v \cdot n > 0$. PVI represents dimensionless time and is computed via

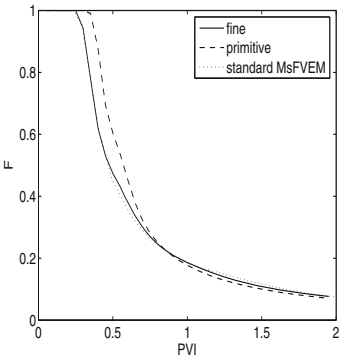

Fig. 2.11. Fractional flow comparison for a permeability field generated using two-point geostatistics.

$$\text{PVI} = \int Qdt/V_p, \tag{2.44}$$

where V_p is the total pore volume of the system and $Q = \int_{\partial\Omega^{\text{out}}} v \cdot nds$ is the total flow rate.

In Figure 2.11, we compare the fractional flows (oil-cut). The dashed line corresponds to the calculations performed using a simple saturation upscaling (no subgrid treatment) where (2.41) is solved with v replaced by the coarse-scale v obtained from MsFVEM. Note that the coarse-scale v is defined as a normal flux, $\int_{\partial K} v \cdot ndl$ along the edge for each coarse-grid block. We call this the primitive model because it ignores the oscillations of v within the coarse-grid block in the computation of S. The dotted line corresponds to the calculations performed by solving the saturation equation on the fine grid using the reconstructed fine-scale velocity field. The fine-scale details of the velocity are reconstructed using the multiscale basis functions. In these simulations, the errors are due to MsFVEM. In the primitive model, the errors are due to both MsFVEM for flow equations (2.40) and the upscaling in the saturation equation (2.41). We observe from this figure that the second approach, where the saturation equation is solved on the fine grid, is very accurate, but the first approach overpredicts the breakthrough time. We note that the second approach contains the errors only due to MsFVEM because the saturation equation is solved on the fine grid. The first approach contains in addition to MsFVEM's errors, the errors due to saturation upscaling which can be large if no subgrid treatment is performed. The saturation snapshots are compared in Figure 2.12. One can observe that there is a very good agreement between the fine-scale saturation and the saturation field obtained using MsFVEM.

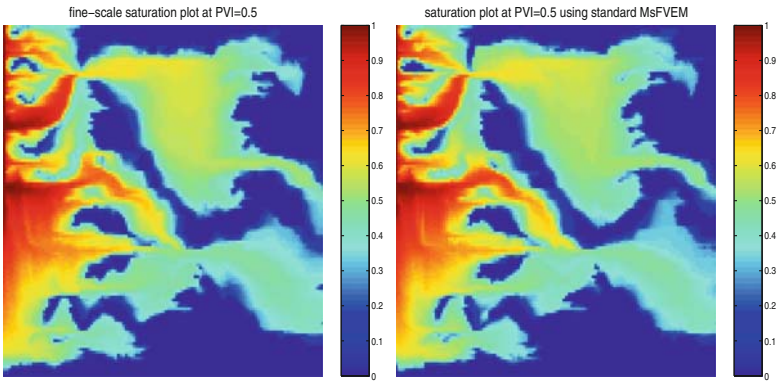

Fig. 2.12. Saturation maps at PVI = 0.5 for fine-scale solution (left figure) and standard MsFVEM (right figure).

2.11 Discussions

In this chapter, we presented an introduction to MsFEMs. We attempted to keep the presentation accessible to a broader audience and avoided some technical details in the presentation. One of the main ingredients of MsFEM is the construction of basis functions. Various approaches can be used to couple multiscale basis functions. This leads to multiscale methods, such as mixed MsFEM, MsFVEM, DG-MsFEM, and so on. Most of the discussions here focus on linear problems and local multiscale basis functions. We have discussed the effects of localized boundary conditions and the approaches to improve them. The relation to some other multiscale methods is discussed.

We would like to note that various multiscale methods are compared numerically in [167]. In particular, the authors in [167] perform comparisons of MsFVM, mixed MsFEM, and variational multiscale methods. Numerical results are performed for various uniform coarse grids and the sensitivities of these approaches with respect to coarse grids are discussed. As we mentioned earlier, one can use general, nonuniform, coarse grids to improve the accuracy of local multiscale methods.

We discussed approximations of basis functions in the presence of strong scale separation. In this case, the basis functions and the elements of the stiffness matrix can be approximated using the solutions in smaller regions (RVE). One can also approximate basis functions by solving the local problems approximately, for example, using approximate analytical solutions [250] for some types of heterogeneities. In [192], the authors propose an approach where the multiscale basis functions are computed inexpensively via multigrid iterations. They show that the obtained method gives nearly the same accuracy on the coarse grid as MsFEM with accurately resolved basis functions.

In this chapter, we did not discuss adaptivity issues that are important for multiscale simulations (see, e.g., [213, 38] for discussions on error estimates and

adaptivity in multiscale simulations). The adaptivity for periodic numerical homogenization within HMM is studied in [214]. In general, one would like to identify the regions where the localization can be performed and the regions where some type of limited global information is needed (see Chapter 4 for the use of limited global information in multiscale simulations). To our best knowledge, such adaptivity issues are not addressed in the literature with a mathematical rigor.

In Section 6.1, we present analysis only for a few multiscale finite element formulations. Our objective is to give the reader a flavor of the analysis, and in particular, stress the subgrid capturing errors. We would like to note that a lot of effort has gone into analyzing multiscale finite element methods. For example, multiscale finite element methods have been analyzed for random homogeneous coefficients [99, 72], for highly oscillatory coefficients with multiple scales [99], for problems with discontinuous coefficients [99], and for various settings of MsFEMs. Our main objective in this book is to give an overview of multiscale finite element methods and present representative cases for the analysis. We believe the results presented in Section 6.1 will help the reader who is interested in the analysis of multiscale finite element methods and, in particular, in estimating subgrid capturing errors.