A MULTISCALE FINITE ELEMENT METHOD FOR ELLIPTIC PROBLEMS IN COMPOSITE MATERIALS AND POROUS MEDIA

THOMAS Y. HOU AND XIAO-HUI WU

APPLIED MATHEMATICS, CALTECH, PASADENA, CA 91125

Abstract. In this paper, we study a multiscale finite element method for solving a class of elliptic problems arising from composite materials and flows in porous media, which contain many spatial scales. The method is designed to efficiently capture the large scale behavior of the solution without resolving all the small scale features. This is accomplished by constructing the multiscale finite element base functions that are adaptive to the local property of the differential operator. Our method is applicable to general multiple-scale problems without restrictive assumptions. The construction of the base functions is fully decoupled from element to element; thus, the method is perfectly parallel and is naturally adapted to massively parallel computers. For the same reason, the method has the ability to handle extremely large degrees of freedom due to highly heterogeneous media, which are intractable by conventional finite element (difference) methods. In contrast to some empirical numerical upscaling methods, the multiscale method is systematic and self-consistent, which makes it easier to analyze. We give a brief analysis of the method, with emphasis on the "resonant sampling" effect. Then, we propose an over-sampling technique to remove the resonance effect. We demonstrate the accuracy and efficiency of our method through extensive numerical experiments, which include problems with random coefficients and problems with continuous scales. Parallel implementation and performance of the method are also addressed.

Key words. multiscale, finite element, homogenization, heterogeneous media, parallel computing

AMS subject classifications. 65F10, 65F30

1. INTRODUCTION

Many problems of fundamental and practical importance have multiple scale solutions. Composite materials, porous media, and turbulent transport in high Reynolds number flows are examples of this type. A complete analysis of these problems is extremely difficult. For example, the difficulty in analyzing groundwater transport is mainly caused by the heterogeneity of subsurface formations spanning over many scales [7]. The heterogeneity is often represented by the multiscale fluctuations in the permeability of the media. For composite materials, the dispersed phases (particles or fibers), which may be randomly distributed in the matrix, give rise to fluctuations in the thermal or electrical conductivity; moreover, the conductivity is usually discontinuous across the phase boundaries. In turbulent transport problems, the convective velocity field fluctuates randomly and contains many scales depending on the Reynolds number of the flow.

A direct numerical solution of the multiple scale problems is difficult even with modern supercomputers. The major difficulty of direct solutions is the scale of computation. For groundwater simulations, it is common to have millions of grid blocks involved, with each block having a dimension of tens of meters, whereas the permeability measured from cores is at a scale of several centimeters [23]. This gives more than 10^5 degrees of freedom per spatial dimension in the computation. Therefore, a tremendous amount of computer memory and CPU time are required, and they can easily exceed the limit of today's computing resources. The situation can be relieved to some degree by parallel computing; however, the size of discrete problem is *not* reduced. The load is merely shared by more processors with more memory. Some recent direct solutions of flow and transport in porous media are reported in [1, 25, 9, 22]. Whenever one can afford to resolve all the small scale features of a physical problem, direct solutions provide quantitative information of the physical processes at all scales. On the other hand, from an engineering perspective, it is often sufficient to predict the macroscopic properties of the multiple-scale systems, such as the effective conductivity, elastic moduli, permeability, and eddy diffusivity. Therefore, it is desirable to develop a method that captures the small scale effect on the large scales, but does not require resolving all the small scale features.

Here, we study a multiscale finite element method (MFEM) for solving partial differential equations with multiscale solutions. The central goal of this approach is to obtain the large scale solutions accurately and efficiently without resolving the small scale details. The main idea is to construct finite element base functions which capture the small scale information within each element. The small scale information is then brought to the large scales through the coupling of the global stiffness matrix. Thus, the effect of small scales on the large scales is correctly captured. In our method, the base functions are constructed from the leading order homogeneous elliptic equation in each element. As a consequence, the base functions are adapted to the local properties of the differential operator. In the case of two-scale periodic structures, Hou, Wu and Cai have proved that the multiscale method indeed converges to the correct solution independent of the small scale in the homogenization limit [21].

In this paper, we continue the study of the multiscale method, with emphasis on problems with continuous scales from composite materials and flows in porous media. Extensive numerical tests are performed on these problems. The error analysis of the method is reviewed briefly for problems with scale separation. The accuracy of our method for problems with continuous scales is then studied numerically. Moreover, we compare our method with traditional finite element (difference) methods as well as existing numerical upscaling methods in terms of operation counts and memory requirement. We give two simple parallel implementations of our method and study their parallel efficiency computationally.

A common difficulty in numerical upscaling methods is that large errors result from the "resonance" between the grid scale and the scales of the continuous problem. This is revealed by our earlier analysis [21]. For the two-scale problem, 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 shows that the boundary layer in the first order corrector seems to be the main source of the resonance effect. By a judicious choice of boundary conditions for the base function, we can eliminate the boundary layer in the first order corrector. This would give a nice conservative difference structure in the discretization, which in turn leads to *cancellation of resonance errors* and gives an improved rate of convergence independent of the small scales in the solution.

Motivated by our earlier analysis [21] mentioned above, here we propose an *over-sampling* method to overcome the difficulty due to scale resonance. The idea is quite simple and easy to implement. Since the boundary layer in the first order corrector is thin, $O(\epsilon)$, we can sample in a domain with size larger than $h + \epsilon$ and use only the interior sampled information to construct the bases (see §3.3). Here, h is the mesh size and ϵ is the small scale in the solution. By doing this, the boundary layer in the larger domain has no influence on the base functions. Now the corresponding first order correctors are free of boundary layers. As a result, we obtain an improved rate of convergence which is independent of the small scale.

From practical considerations, this improvement is crucial. For problems with many scales or continuous scales, it is inevitable to have the mesh size h coincide with one of the physical scales. Without this improvement, we can not guarantee that our method converges completely independent of the small scale features in the solution. It is also important that our over-sampling technique does not rely on the homogenization theory (like solving a cell problem), although the homogenization theory helps reveal the cause of the problem. This makes it possible to generalize our method to problems with continuous scales. We will demonstrate through extensive numerical experiments that this simple technique is very effective for a wide range of applications, including problems with random coefficients and problems with continuous scales.

In practical computations, a large amount of overhead time comes from constructing the base functions. These multiscale base functions are constructed numerically, except for certain special cases. Since the base functions are independent of each other, they can be constructed independently and this can be done perfectly in parallel. This greatly reduces the overhead time in constructing these bases. On a sequential machine, the operation count of our method is about twice that of a conventional finite element method (FEM) for a 2-D problem. The difference is reduced significantly for a massively parallel computer. For example, running on 256 processors, our method only spends 9% more CPU time than a FEM using 1024×1024 linear elements (see §4.6).

Another advantage of our method is its ability to reduce the size of a large scale computation. This offers a big saving in computer memory. For example, let N to be the number of elements in each spatial direction, and M be the number of subcell elements in each direction for solving the base functions. Then there are total $(MN)^n$ (n is the dimension) elements at the fine grid level. For a traditional FEM, the computer memory needed for solving the problem on the fine grid is $O(M^n N^n)$. In contrast, MFEM requires only $O(M^n + N^n)$ amount of memory. If $M = 32$ in a 2-D problem, then traditional FEM needs about 1000 times more memory than MFEM.

Since we need to use an additional grid to compute the base function numerically, it makes sense to compare our multiscale FEM with a traditional FEM at the subcell grid, $h_s = h/M$. Note that the multiscale FEM only captures the solution at the coarse grid h, while a traditional FEM tries to resolve the solution at the fine grid $h_s = h/M$. Our extensive numerical experiments demonstrate that the accuracy of our multiscale FEM on the coarse grid h is comparable to that of FEM on the fine grid. In some cases, MFEM is even more accurate than FEM (see §4.3 and 4.4).

At this point, we would like to emphasize that the purpose of our method is to solve practical problems which are too large to handle by direct methods on given computing resources. Our method gives a systematic and self-consistent approach to capture the large scale solution correctly without resolving the small scale details and without resorting to closure arguments. We show that at a reasonable cost, the multiscale FEM has the ability to solve very large scale practical problems with accuracy comparable to the corresponding direct simulations at the fine grid. This gives hope to solve some large scale computational problems that are otherwise intractable using direct methods.

It should be mentioned that many numerical methods have been developed with goals similar to ours. These include methods based on the homogenization theory (cf. [14, 10]), and some upscaling methods based on simple physical and/or mathematical motivations (cf. [12, 23]). The methods based on the homogenization theory have been successfully applied to determining the effective conductivity and permeability of certain composite materials and porous media [14, 10]. However, their range of applications is usually limited by restrictive assumptions on the media, such scale separation and periodicity [8]. As discussed in §4.2, they are also expensive to use for solving problems with many separate scales since the cost of computation grows exponentially with the number of scales. But for the multiscale method, the number of scales is irrelevant to the computational cost. The upscaling methods are more general and have been applied to problems with random coefficients with partial success (cf. [12, 23]). But the design principle is strongly motivated by the homogenization theory for periodic structures. Their applications to non-periodic structures are not always guaranteed to work.

There has also been success in achieving numerical homogenization for some semilinear hyperbolic systems, the incompressible Euler equations, and 1-D elliptic problems using the sampling technique, see e.g. [17, 15, 2]. This technique has its own limitations. Its application to general 2-D elliptic problems is still not satisfactory. For fully random media, statistical theory and renormalization group theory have been used to obtain the effective properties. However, these methods usually become difficult to apply when the integral scale of correlation is large (ref. [23] and references therein). Moreover, certain simplifying assumptions in the underlying physics are usually made in order to obtain a closure of the effective equations. In comparison, such a closure problem is not present in the multiscale method.

We remark that the idea of using base functions governed by the differential equations has been applied to convection-diffusion equation with boundary layers (see, e.g., [6] and references therein). With a motivation different from ours, Babuska et al. applied a similar idea to 1-D problems [5] and to a special class of 2-D problems with the coefficient varying locally in one direction [4]. However, most of these methods are based on the special property of the harmonic average in one-dimensional elliptic problems. As indicated by our convergence analysis, there is a fundamental difference between one-dimensional problems and genuinely multi-dimensional problems. Special complications such as the resonance between the mesh scale and the physical scale never occur in the corresponding 1-D problems.

This paper is organized as follows. The formulation of the 2-D multiple-scale elliptic problem and the multiscale finite element method are given in the next section. In Section 3, we present the rationale behind the method, including a brief review of the homogenization theory and convergence analysis. The resonance effect is analyzed and the over-sampling technique is proposed. More detailed numerical analysis of the method is given in a separate paper [21]. The numerical implementation of the method, its convergence, and parallel performance are studied in Section 4. Section 5 contains the application of the multiscale method to more practical problems in composite materials and porous media flows, including steady conduction through fiber composites and flows through random porous media with normal and fractal porosity distributions. Using these examples, we show the adaptability of the method, its ability to solve large practical problems, and its accuracy for general problems. Section 6 is reserved for some concluding remarks and discussion of future work.

2. FORMULATIONS

In this section, we introduce the elliptic problem and the multiscale method. First, we state some notations and conventions to be used in the paper. In the following, the Einstein summation convention is used: summation is taken over repeated indices. Some notations of functional spaces will be used occasionally for the convenience of expressing the formulation and some relevant analytical estimates about the multiscale method. $L^2(\Omega)$ denotes the space of square integrable functions defined in domain Ω. We use $L^2(\Omega)$ based Sobolev spaces $H^k(\Omega)$ equipped with norms and seminorms given by

$$\|u\|_{k,\Omega}^2 = \int_\Omega \sum_{|\alpha|\le k} |D^\alpha u|^2, \qquad |u|_{k,\Omega}^2 = \int_\Omega \sum_{|\alpha|=k} |D^\alpha u|^2,$$

where $D^\alpha u$ denotes the α-th order mixed derivatives of u. $H_0^1(\Omega)$ consists of those functions in $H^1(\Omega)$ that vanish on $\partial\Omega$.

2.1. Governing Equations and the Multiscale Finite Element Method

We consider solving the second order elliptic equation:

$$-\nabla \cdot a(\mathbf{x})\nabla u = f \qquad \text{in } \Omega, \tag{2.1}$$

where $a(\mathbf{x}) = (a_{ij}(\mathbf{x}))$ is the conductivity tensor and is assumed to be symmetric and positive definite with upper and lower bounds. In the context of porous flows, Eq. (2.1) is the pressure equation for single phase steady flow through a porous medium. Correspondingly, a is the ratio of the permeability tensor κ and the fluid viscosity μ, and u represents the pressure. The steady velocity field is related to the pressure through Darcy's law:

$$\mathbf{q} = -\frac{1}{\mu}\kappa\nabla u = -a\nabla u. \tag{2.2}$$

In this paper, we assume $\mu = 1$ for convenience. Eq. (2.1) is also the equation of steady state heat (electrical) conduction through a composite material, with a and u interpreted as the thermal (electric) conductivity and temperature (electric potential). In practice, a may be random or highly oscillatory; thus the solution of (2.1) displays a multiple scale structure. Since for the transient problem the main difficulty is the same as that for the steady state problem, i.e., the multiple scales in the solution, we only consider solving the steady problem here. The multiscale method, however, can be easily extended to solve the transient problems.

To simplify the presentation of the finite element formulation, we assume $u = 0$ on $\partial\Omega$ and that the solution domain is a unit square $\Omega = (0,1) \times (0,1)$. The variational problem of (2.1) is to seek $u \in H_0^1(\Omega)$ such that

$$a(u,v) = f(v), \qquad \forall v \in H_0^1(\Omega), \tag{2.3}$$

where

$$a(u,v) = \int_\Omega a_{ij}\frac{\partial v}{\partial x_i}\frac{\partial u}{\partial x_j}d\mathbf{x} \quad \text{and} \quad f(v) = \int_\Omega fv d\mathbf{x}.$$

A finite element method is obtained by restricting the weak formulation (2.3) to a finite dimensional subspace of $H_0^1(\Omega)$. For $0 < h \le 1$, let $\mathcal{K}^h$ be a partition of Ω by a collection of rectangles K with diameter $\le h$, which is defined by an axi-parallel rectangular mesh (Fig. 2.1). In each element $K \in \mathcal{K}^h$, we define a set of nodal basis $\{\phi_K^i, i = 1, \ldots, d\}$ with d being the number of nodes of the element. The subscript K will be neglected when bases in one element are considered. In our multiscale method, ϕ^i satisfies

$$\nabla \cdot a(\mathbf{x})\nabla\phi^i = 0 \qquad \text{in } K \in \mathcal{K}^h. \tag{2.4}$$

Let $\mathbf{x}_j \in \overline{K}$ $(j = 1, \ldots, d)$ be the nodal points of K. As usual, we require $\phi^i(\mathbf{x}_j) = \delta_{ij}$. One needs to specify the boundary condition of ϕ^i to make (2.4) a well-posed problem (see below). For now, we assume that the base functions are continuous across the boundaries of the elements, so that

$$V^h = \operatorname{span}\{\phi_K^i : i = 1, \ldots, d; K \in \mathcal{K}^h\} \subset H_0^1(\Omega).$$

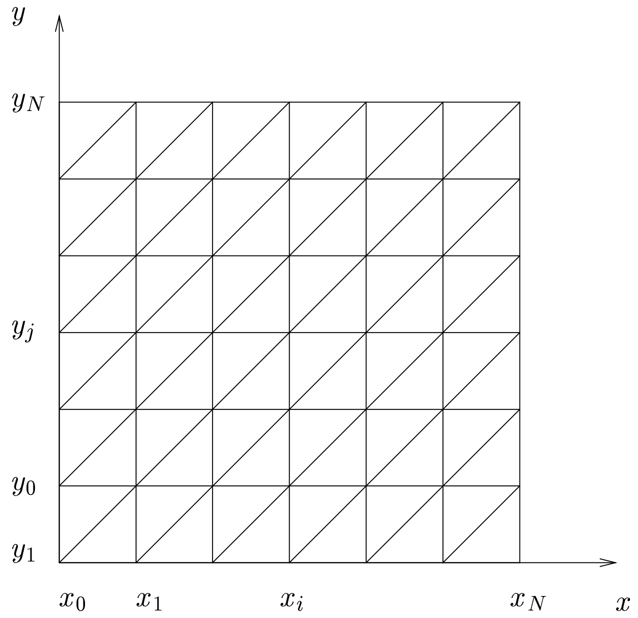

FIG. 2.1. *Rectangular mesh with triangulation.*

In the following, we study the approximate solution of (2.3) in V^h, i.e., $u^h \in V^h$ such that

$$a(u^h, v) = f(v), \qquad \forall v \in V^h. \tag{2.5}$$

Note that this formulation of the multiscale method is not restricted to rectangular elements. It can also be applied to triangular elements (see Fig. 2.1) which are more flexible in modeling complicated geometries.

2.2. The Boundary Condition of Base Functions

The important role of the boundary condition of the base functions is obvious since the base functions satisfy the homogeneous equation (2.4). We will see later that a good choice of the boundary condition can significantly improve the accuracy of the multiscale method. In fact, the boundary condition determines how well the local property of the operator is sampled into the base functions (see §3). Here, we describe two methods of imposing the boundary condition, which are easy to implement and to analyze.

Denote $\mu^i = \phi^i|_{\partial K}$. One choice is to let μ^i vary linearly along ∂K, just as in the standard bilinear (linear) base functions. Another more appealing approach is to choose μ^i to be the solution of some reduced elliptic problems on each side of ∂K. The reduced problems are obtained from (2.4) by deleting terms with partial derivatives in the direction normal to ∂K and having the coordinate normal to ∂K as a parameter. It is clear that the reduced problems are of the same form as (2.4). When a is separable in space, i.e., $a(\mathbf{x}) = a_1(x)a_2(y)$, ϕ^i can be computed analytically from the tensor product of μ^i along Γ_{i-1} and Γ_i (note that $\Gamma_0 \equiv \Gamma_4$, see Fig. 2.2). Furthermore, it can be shown that this boundary condition is optimum for the space-separable problems.

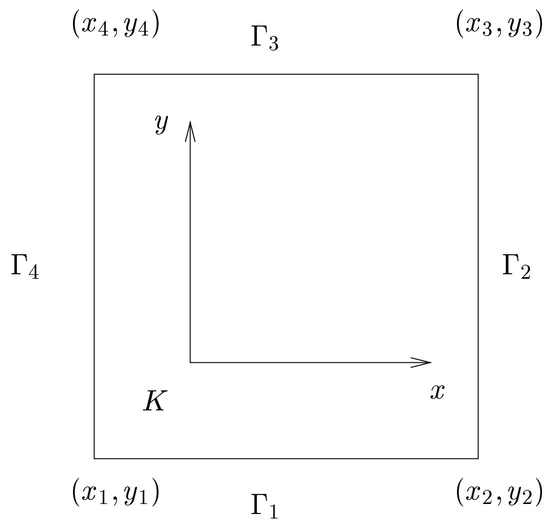

FIG. 2.2.

To be more specific, consider an element $K \in \mathcal{K}^h$ with nodal points $\mathbf{x}_i = (x_i, y_i)$ $(i = 1, \dots, d)$, which are labeled counterclockwise starting from the lower left corner (Fig. 2.2). On Γ_1 and Γ_3, we have $\mu^i = \mu^i(x)$ and

$$\frac{\partial}{\partial x} a_\mu(x) \frac{\partial \mu^i(x)}{\partial x} = 0, \tag{2.6}$$

where $a_\mu(x) = a_{11}|_{\Gamma_1}$ and $a_{11}|_{\Gamma_3}$, respectively. Note that a_μ is bounded from above and below by positive constants. Similarly, on Γ_2 and Γ_4, we have $\mu^i = \mu^i(y)$ and

$$\frac{\partial}{\partial y} a_\mu(y) \frac{\partial \mu^i(y)}{\partial y} = 0$$

with $a_\mu(y) = a_{22}|_{\Gamma_2}$ and $a_{22}|_{\Gamma_4}$, respectively. The boundary condition of these 1-D elliptic equations is given by $\mu^i(\mathbf{x}_j) = \delta_{ij}$. The equations can be solved analytically. For example, on Γ_1 we have

$$\mu^1(x) = \int_x^{x_2} \frac{dt}{a_\mu(t)} \Big/ \int_{x_1}^{x_2} \frac{dt}{a_\mu(t)}. \tag{2.7}$$

If a_μ is a constant, then $\mu^1(x) = (x_2 - x)/(x_2 - x_1)$ is linear. In general, μ^is are oscillatory due to the oscillations in a_μ. One may verify that using the above boundary conditions, the base functions are continuous across ∂K. Also, with both types of boundary condition, one has

$$\sum_{i=1}^{d} \phi_K^i = 1, \qquad \forall K \in \mathcal{K}^h. \tag{2.8}$$

Thus, the constant functions belong to V^h. Later, we see that this property is useful in discrete error cancellations. The generalization of the reduced problems, e.g.,(2.6), to more general elements, such as the triangular elements, is straightforward.

2.3. Some General Remarks

The multiscale method formulated above is designed to capture the large scale solutions. Unlike existing numerical upscaling methods, our method is consistent with the traditional finite element method in a well-resolved computation. It is proved in [21] that the multiscale method gives the same rate of convergence as the linear finite element method when the small scales are well resolved, $h << \epsilon$. In particular, when the coefficient is a diagonal constant matrix, the base functions constructed from (2.4) are nothing but the usual bilinear (linear) base functions. When h does not resolve the small scales, the multiscale method and the traditional finite element method behave very differently. It is easy to show that the traditional finite element methods do not converge to the correct solution. By contrast, the multiscale method captures the correct large scale solutions.

As indicated by our analysis and numerical experiments in [21], the boundary condition of the base functions can have a big influence on the accuracy of the multiscale method. From our computational experience, we found that the oscillatory boundary condition for the base functions in general leads to better accuracy than the linear boundary condition. However, the multiscale method in general may fail to converge when the mesh scale is close to the physical small scale due to a resonance between these two scales. For the two-scale problem, the error due to the resonance manifests as a ratio between the wavelength of the small scale oscillation and the grid size. Motivated by our earlier analysis [21], we propose in §4.4 an *over-sampling* method to overcome the difficulty due to scale resonance.

3. THEORETICAL BACKGROUND

We use a model elliptic problem to provide some insights to the multiscale method and the rationale behind the over-sampling scheme. Here, we only briefly outline the analysis. The main concern is how to remove the "resonance" effect.

3.1. The Model Problem and Homogenization

In the model problem, the coefficient is chosen as $a = a(\mathbf{x}/\epsilon)$, where ϵ is a small parameter, characterizing the small scale of the problem. We assume $a(\mathbf{y})$ to be periodic in Y and smooth. We denote the volume average over Y as $\langle\cdot\rangle = \frac{1}{|Y|}\int_Y \cdot\, d\mathbf{y}$. As in §2, we assume $u = 0$ on $\partial\Omega$.

By the homogenization theory [8], the solution of (2.1) has an asymptotic expansion, i.e.,

$$u = u_0(\mathbf{x}) + \epsilon u_1(\mathbf{x}, \mathbf{y}) - \epsilon\theta_\epsilon + O(\epsilon^2) \tag{3.1}$$

where $\mathbf{y} = \mathbf{x}/\epsilon$ is the fast variable. Here, u_0 is the solution of the homogenized equation

$$\nabla \cdot a^* \nabla u_0 = f \quad \text{in } \Omega, \qquad u_0 = 0 \quad \text{on } \partial\Omega, \tag{3.2}$$

a^* is the constant effective coefficient, given by

$${a^*}_{ij} = \langle a_{ik}(\mathbf{y})(\delta_{kj} - \frac{\partial}{\partial y_k}\chi^j)\rangle, \tag{3.3}$$

and χ^j is the periodic solution of

$$\nabla_{\mathbf{y}} \cdot a(\mathbf{y}) \nabla_{\mathbf{y}} \chi^j = \frac{\partial}{\partial y_i} a_{ij}(\mathbf{y}) \tag{3.4}$$

with zero mean, i.e., $\langle \chi^j \rangle = 0$. It is proved in [8] that a^* is symmetric and positive definite. Moreover, we have

$$u_1(\mathbf{x}, \mathbf{y}) = -\chi^j \frac{\partial u_0}{\partial x_j}. \tag{3.5}$$

Since in general $u_1 \neq 0$ on $\partial\Omega$, the boundary condition $u|_{\partial\Omega} = 0$ is enforced through the first order correction term θ_ϵ, which is given by

$$\nabla \cdot a(\mathbf{x}/\epsilon)\nabla\theta_\epsilon = 0 \quad \text{in } \Omega \quad \text{and} \quad \theta_\epsilon = u_1(\mathbf{x}, \mathbf{x}/\epsilon) \quad \text{on } \partial\Omega. \tag{3.6}$$

The asymptotic expansion (3.1) has been rigorously justified in [8]. Under certain smoothness conditions, one can also obtain point-wise convergence of u to u_0 as $\epsilon \to 0$. The conditions can be weakened if the convergence is considered in the $L^2(\Omega)$ space.

As mentioned in §1, some numerical upscaling methods are directly based on (3.4), (3.3), and (3.2); see, e.g., [14, 10]. We use these results only for the convenience of analysis. Indeed, the asymptotic structure (3.1) is used to reveal the subtle details of the multiscale method and obtain sharp error estimates [21]. Without using this structure, the conventional finite element analysis does not give correct answers. An extension of the convergence analysis to the multiple scale problems is given in [16].

3.2. Error Estimates and the Resonance Effect

In [21], we prove that the multiscale method converges to the correct homogenized solution in the $\epsilon \to 0$ limit. This can be summarized from the following estimate:

THEOREM 3.1. *Let u and u^h be the solutions of* (2.1) *and* (2.5), *respectively. Then there exist positive constants C_1 and C_2, independent of ϵ and h, such that*

$$\|u - u^h\|_{1,\Omega} \leq C_1 h \|f\|_{0,\Omega} + C_2(\epsilon/h)^{\frac{1}{2}} \qquad (\epsilon < h). \tag{3.7}$$

The key to (3.7) is that the base functions defined by (2.4) have the same asymptotic structure as that of u, i.e.,

$$\phi^i = \phi_0^i + \epsilon\phi_1^i - \epsilon\theta^i + \cdots \quad (i = 1, \dots, d), \tag{3.8}$$

where ϕ_0^i, ϕ_1^i, and θ^i are defined similarly as u_0, u_1, and θ_ϵ, respectively. We note that if a^* is diagonal (i.e., isotropic), then ϕ_0^i becomes the usual bilinear base function. We would like to point out that applying the conventional finite element analysis to our multiscale method gives an overly pessimistic estimate $O(h/\epsilon)$ in the H^1 norm, which is only useful for $h \ll \epsilon$. It is important that we obtain an estimate in the form of ϵ/h for our multiscale method. This shows that our method converges to the correct homogenized solution in the limit as $\epsilon \to 0$. This property is not shared by the conventional finite element methods with polynomial bases, since small scale information is averaged out incorrectly.

The L^2-norm error estimate can be obtained from (3.7) by using the standard finite element analysis. However, again, the error is over-estimated. In [21], it is shown that

$$\|u - u^h\|_{0,\Omega} \leq C_1 h^2 \|f\|_{0,\Omega} + C_2\epsilon + C_3\|u^h - u_0^h\|_{l^2(\Omega)}$$

where u_0^h is the solution of (3.2) using ϕ_0^is as the base functions, and $C_i > 0$ $(i = 1, 2, 3)$ are constants independent of ϵ and h. The discrete l^2 norm $\|\cdot\|_{l^2(\Omega)}$ is given by

$$\|u^h\|_{l^2(\Omega)} = \left(\sum_{i\in\mathcal{N}} u^h(\mathbf{x}_i)^2 h^2\right)^{\frac{1}{2}},$$

where $\mathcal{N}$ is the set of indices of all nodal points on the mesh. We will see below that in general, $\|u^h - u_0^h\|_{l^2(\Omega)} = O(\epsilon/h)$. Thus, we have

$$\|u - u^h\|_{0,\Omega} = O(h^2 + \epsilon/h).$$

It is now clear that when $h \sim \epsilon$, the multiscale method attains large error in both H^1 and L^2 norms. This is what we call the *resonance* effect between the grid scale (h) and the small scale (ϵ) of the problem. This estimate reflects the intrinsic scale interaction between the two scales in the *discrete* problem. Our extensive numerical experiments confirm that this estimate is indeed generic and sharp. It should be pointed out that the estimate only provides the rate of convergence; the actual numerical error of the multiscale method in the resonant regime can still be small due to a small error constant in $O(\epsilon/h)$. This is indeed the case as shown by our numerical tests in [21]. However, by removing the resonance effect, we can greatly improve the accuracy and the convergence rate. Such an improvement is especially important for problems with continuous scales, because there is always a scale of the problem that coincides with the grid scale and hence the resonance effect cannot be avoided by varying h. In §3.3, an over-sampling method is proposed to overcome this difficulty.

The mechanism of the resonance effect can be understood from a discrete error analysis [21]. For convenience, we outline the analysis here without giving the details of the derivation. We derive the $O(\epsilon/h)$ estimate for the l^2-norm convergence and illustrate the difficulty in improving the convergence rate.

Let U^h and U_0^h denote the nodal point values of u^h and u_0^h, respectively. The linear system of equations for U^h is

$$A^h U^h = f^h, \tag{3.9}$$

where A^h and f^h are obtained from $a(u^h, v)$ and $f(v)$ by using $v = \phi^i$ for $i \in \mathcal{N}$. Similarly, for U_0^h one has

$$A_0^h U_0^h = f_0^h, \tag{3.10}$$

where A_0^h and f_0^h are obtained by applying $v = \phi_0^i$ $(i \in \mathcal{N})$ to $a^*(u_0^h, v) = f(v)$ with

$$a^*(u_0^h, v) = \int_\Omega a_{ij}^* \frac{\partial v}{\partial x_i} \frac{\partial u_0^h}{\partial x_j} d\mathbf{x}.$$

By using (3.8), it can be shown that

$$A^h = A_0^h + \frac{\epsilon}{h} A_1^h + O(\frac{\epsilon^2}{h^2}) \qquad \text{and} \qquad f^h = f_0^h + \frac{\epsilon}{h} f_1^h + O(\frac{\epsilon^2}{h^2}), \tag{3.11}$$

where the elements of matrix A_1^h and vector f_1^h are $O(1)$ and $O(h^2)$, respectively. The expansion of A^h indicates that the homogenized differential operator is captured at the discrete level by the multiscale base functions. It follows immediately that U^h can be expanded as

$$U^h = U_0^h + \frac{\epsilon}{h} U_1^h + \cdots,$$

thus U^h converging to U_0^h as $\epsilon \to 0$. To obtain the convergence rate, it remains to determine the order of U_1^h. Substituting the expansions of A^h, f^h, and U^h into (3.9), we obtain

$$A_0^h U_1^h = f_1^h - A_1^h U_0^h. \tag{3.12}$$

Letting $G^h = (A_0^h)^{-1}$, we have $U_1^h = G^h f_1^h - G^h A_1^h U_0^h$. Note that G^h consists of the nodal values of the finite element projection of $G(\mathbf{x}, \boldsymbol{\xi})$, the continuous Green's function for the homogenized equation (3.2). The properties of G^h have been studied in [19, 24]. It turns out that G^h is similar to G, which has a $\log|\mathbf{x} - \boldsymbol{\xi}|$ type of singularity. Like the continuous Green's function, G^h is absolutely summable over the whole domain. Thus, by direct summation one has $G^h f_1^h = O(1)$. However, the direct summation gives $G^h A_1^h U_0^h = O(1/h^2)$, which is an over estimate. By (2.8) and the symmetry of A_1^h, one can write $A_1^h U_0^h$ in a *conservative* form [21]:

$$(A_1^h U_0^h)_{ij} = \sum_{s=1}^{4} (D_s^+ B_{ij}^s D_s^-) U_{0ij}^h, \tag{3.13}$$

where (i, j) is the 2-D index for grid points (Fig. 2.1), and B_{ij}^s obtained from A_1^h are the weights for the stencil. Here, D_s^+ and D_s^- are the forward and backward difference operators in the horizontal, the vertical, and the two diagonal directions for $s = 1$ to 4, respectively. For example, we have $D_1^+ U_{ij}^h = U_{i+1j}^h - U_{ij}^h$ and $D_3^- U_{ij}^h = U_{ij}^h - U_{i-1j-1}^h$. For further details see Appendix B of [21]. We note that $D_s^- U_0^h = O(h)$ since U_0^h is an $O(h^2)$ approximation of the smooth function u_0. Now, consider

$$\sum_{i,j=1}^{N-1} G_{lm,ij}^h (A_1^h U_0^h)_{ij} \qquad (l, m = 1, \ldots, N-1).$$

Note that the indices of the matrix entry G_{pq}^h have been translated into the 2-D indices $p = (l, m)$ and $q = (i, j)$ for the nodal points. Observe that by using *summation by parts*, one can transfer the action of D_s^+ onto G^h, which gives $D_s^- G^h$. As an example, consider applying G^h to the first term ($s = 1$) of the sum in (3.13). Neglecting indices l and m, we have

$$\sum_{i,j=1}^{N-1} G_{ij}^h D_1^+ B_{ij}^1 D_1^- (U_0^h)_{ij} = -\sum_{j=1}^{N-1} \sum_{i=1}^{N} (D_1^- G_{ij}^h) B_{ij}^1 (D_1^- (U_0^h)_{ij}) \qquad (G_{0j}^h = G_{Nj}^h = 0).$$

We note that the divided difference $D_s^- G^h / h$ is absolutely summable [21]. It follows that $G^h A_1^h U_0^h = O(1)$ and hence $U_1^h = O(1)$. Thus, we obtain

$$U^h - U_0^h = O(\epsilon/h).$$

The derivation shows that the error cancellation is mainly due to the difference structures in $A_1^h U_0^h$ given by (3.13). Clearly, the estimate of $U^h - U_0^h$ could be further improved by using summation by parts again *if* B^s and f_1^h can be written in difference forms, e.g.,

$$B_{ij}^s = D_1^- C_{ij}^s + D_2^- D_{ij}^s \ (s = 1, \ldots, 4) \quad \text{and} \quad f_{1ij}^h = D_1^- E_{ij} + D_2^- F_{ij},$$

where C^s, D^s, E, and F are uniquely defined on the nodal points. Then, we would have for $\epsilon < h$, $\|u - u^h\|_{0,\Omega} = O(h^2 + \epsilon|\log(h)|)$, independent of ϵ. In this case, the interaction between the h and ϵ scales is very weak and hence the resonance effect disappears. Note that the factor $\log(h)$ comes from the sum of terms with second order divided differences of G^h, e.g., $D_2^+ D_1^- G^h/h^2$. The singularity in the second order derivatives of the discrete Green function, which is similar to its continuous counterpart, contributes to the factor $\log(h)$. See [21] for further details of the derivation.

The method of exploring the difference forms in B^s and f_1^h has been given in [21]. The idea is to recast the volume integrals in B^s and f_1^h into boundary integrals. Then, the opposite directions of outward normal vectors of two neighboring elements lead to the difference structures, provided that the integrands of the boundary integrals are continuous at the interfaces of elements. In this regard, the triangular element is much easier to analyze since ϕ_0^is are always linear. In comparison, ϕ_0^is are in general some unknown functions for rectangular elements. Therefore, in the following we give an analysis for the triangular elements, e.g., the triangulation in Fig. 2.1.

We find that f_1^h can indeed be written in a difference form. However B^s cannot be written in difference forms due to the following boundary integral

$$B_\theta = \epsilon' \int_{\partial K'} \theta^k n_i a_{ij} \frac{\partial \theta^l}{\partial x'_j} ds' \qquad (k, l = 1, \dots, d),$$

where θ^k $(k = 1, \dots, d)$ is the first order corrector in (3.8) and the prime indicates that the variable and the domain have been rescaled by h, i.e., $\epsilon' = \epsilon/h$ and $\mathbf{x}' = \mathbf{x}/h$ (see Appendix B of [21]). Thus, we identify θ^k as the main source of the resonance effect.

To further understand the problem, let us examine θ^k more closely. Since θ^k satisfies the homogeneous equation (3.6) in the interior and is highly oscillatory on the boundary, it can be shown that θ^k has a special solution structure. Let

$$P_\epsilon(\mathbf{x}', \boldsymbol{\xi}') = \partial G_\epsilon(\mathbf{x}', \boldsymbol{\xi}')/\partial n_{\boldsymbol{\xi}'} \qquad (\mathbf{x}' \in K', \ \boldsymbol{\xi}' \in \partial K')$$

be the Poisson kernel for L'_ϵ, where $G_\epsilon(\mathbf{x}', \boldsymbol{\xi}')$ is the Green's function of the Dirichlet problem for θ^k. Furthermore, we assume that $\theta^k = g(\mathbf{x}'/\epsilon)$ on the boundary $\partial K'$. Then we have

$$\theta^k(\mathbf{x}') = \int_{\partial K'} P_\epsilon(\mathbf{x}', \boldsymbol{\xi}') g(\boldsymbol{\xi}'/\epsilon') d\boldsymbol{\xi}'.$$

It has been shown in [3] that to the leading order P_ϵ can be approximated by a smooth kernel $d(\mathbf{x}')/|\mathbf{x}' - \boldsymbol{\xi}'|^2$, where $d(\mathbf{x}')$ is the distance function from $\mathbf{x}'$ to $\partial K'$. Thus, the integral expression of $\theta^k(\mathbf{x}')$ shows that near $\partial K'$ there exists a boundary layer with a thickness of $O(\epsilon')$, in which θ^k has $O(1)$ oscillations (see Fig. 4.1). Away from the boundary layer, the oscillation is only $O(\epsilon')$. Therefore, $\partial\theta^k/\partial x'_j$ is $O(1/\epsilon')$ near $\partial K'$ but is $O(1)$ away from the boundary.

In general, it is impossible to express B_θ in difference forms. However, if we could remove the boundary layer of θ^k so that $\partial\theta^k/\partial x'_j = O(1)$ on $\partial K'$, then B_θ would become $O(\epsilon/h)$ and would not influence the leading order convergence rate. We note that the structure of θ^k is solely determined by its boundary condition, which in turn is determined by the boundary condition of ϕ^k. Therefore, a judicious choice of μ^k may remove the boundary layer of θ^k. We will investigate this idea in the next subsection.

3.3. The Over-Sampling Method

From the above discussion, we see that the first order corrector θ^k has a boundary layer structure when its boundary condition on ∂K has a high frequency oscillation with $O(1)$ amplitude. Thus, in order to further error cancellations in the discrete system, we would like to eliminate the boundary layer structure by choosing a proper boundary condition for the base function ϕ^k. This will give rise to a conservative difference form in the coefficient B^s, which leads to an improved rate of convergence for the multiscale method, independent of the mesh scale. Such a boundary condition does exist, e.g., we may set $\phi^k = \phi_0^k + \epsilon\phi_1^k$ on ∂K (see (3.8)), which enforces $\theta^k = 0$ in K. We do not advocate such an approach since ϕ_1^k needs to be solved from the cell problem which is in general not available except for periodic structures. In the special case when a is diagonal and separable in 2-D, the base functions can be constructed from the tensor products of the corresponding 1-D bases. This construction corresponds to using the oscillatory μ^k (see §2.2) as the boundary condition for ϕ^k. In this case, it is easy to show that the corrector θ^k does not have a boundary layer. This is a special example of obtaining the appropriate boundary condition without solving the cell problem.

The above ideal boundary condition, which makes $\theta^k \equiv 0$ in K, demonstrates an important point: the boundary condition of ϕ^k should match the oscillation of ϕ_1^k (or χ^j) on ∂K. Since the information contained in χ^j is *two-dimensional*, it is difficult if not impossible to extract this information using a 1-D procedure, such as those given in §2.2.

Motivated by the analysis of §3.3, we propose a simple strategy to overcome the influence of the boundary layer. Since the boundary layer of θ^k is thin, only of $O(\epsilon)$ (in the original scale), we can sample in a domain with size larger than $h+\epsilon$ and use only the interior information to construct the base functions. In this way, the boundary layers in the "sampling" domain have no influence on the base functions. Any reasonable boundary condition can be imposed on the boundary of that domain.

Specifically, we construct the base functions for a sampling element $S \supset K$ with $\mathrm{diam}(S) = H > h+\epsilon$ (see Fig 3.1). Denote these temporary base functions as ψ^i $(i = 1, \dots, d)$. We then construct the actual base functions from the linear combination of ψ^js, i.e.,

$$\phi^i = \sum_{j=1}^{d} c_{ij}\psi^j \qquad (i = 1, \dots, d),$$

where c_{ij} are the constants determined by the condition $\phi^i(\mathbf{x}_j) = \delta_{ij}$. Thus, $(c_{ij}) = \Psi^{-1}$, where matrix Ψ is given by $(\psi^i(\mathbf{x}_j))$. Below, we show that the resulting base functions have expansions with a structure very close to that of (3.8); thus previous analysis can be used to study the new base functions. We will use ψ to denote the vector formed by ψ^i $(i = 1, \dots, d)$. Similar notations apply to other variables with superscripts.

Since $\nabla \cdot a(\mathbf{x}/\epsilon)\nabla\psi = 0$, we can expand ψ as

$$\psi = \psi_0 + \epsilon\psi_1 - \epsilon\theta_\psi + O(\epsilon^2), \tag{3.14}$$

where ψ_0, ψ_1, and θ_ψ are defined similarly as in (3.8) in domain S. Correspondingly, we have the matrix expansion

$$\Psi = \Psi_0 + \epsilon\Psi_1 - \epsilon\Theta + O(\epsilon^2).$$

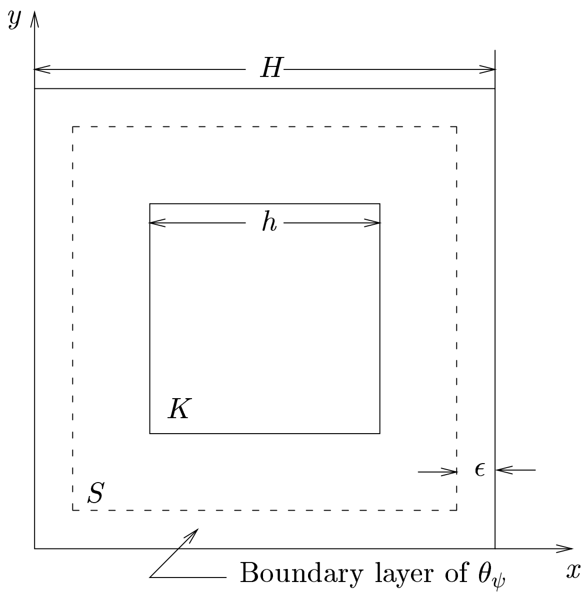

FIG. 3.1. *Adaptive base construction using samples from larger domain to avoid the boundary effect*

The inverse of Ψ may be formally expanded as

$$\begin{aligned}\Psi^{-1} &= (\Psi_0 + \epsilon(\Psi_1 - \Theta) + \cdots)^{-1} \\ &= [\Psi_0(I + \epsilon\Psi_0^{-1}(\Psi_1 - \Theta) + \cdots)]^{-1} \\ &= \Psi_0^{-1} - \epsilon\Psi_0^{-1}(\Psi_1 - \Theta)\Psi_0^{-1} + O(\epsilon^2).\end{aligned} \tag{3.15}$$

Thus, if Ψ_0^{-1} exists and $\|\epsilon\Psi_0^{-1}(\Psi_1 - \Theta)\|$ is sufficiently small, then the expansion converges and Ψ^{-1} exists. In general, the existence of Ψ_0^{-1} is unknown, but since ψ_0^i are close to the bilinear base functions for rectangular elements which are linearly independent, Ψ_0^{-1} exists under fairly weak conditions. For triangular elements, the existence of Ψ_0^{-1} is guaranteed since ψ_0^i are the linear bases. Moreover, it can be seen that $\|\Psi_0^{-1}\| \sim H/h$ and $\|\Psi_1 - \Theta\| \sim 1/H$. Hence the convergence criterion for (3.15) is ϵ/h being small. This is independent of H. Substituting (3.14) and (3.15) into $\phi = \Psi^{-1}\psi$ yields

$$\phi = \Psi_0^{-1}\psi_0 + \epsilon\Psi_0^{-1}\psi_1 - \epsilon\Psi_0^{-1}\theta_\psi - \epsilon\Psi_0^{-1}(\Psi_1 - \Theta)\Psi_0^{-1}\psi_0 + O(\epsilon^2).$$

Define $\phi_0 = \Psi_0^{-1}\psi_0$. We have

$$\phi = \phi_0 + \epsilon\phi_1 - \epsilon\Psi_0^{-1}\theta_\psi - \epsilon\Psi_0^{-1}(\Psi_1 - \Theta)\phi_0 + O(\epsilon^2), \tag{3.16}$$

where ϕ_1 is related to ϕ_0 by (3.5). Note that if ψ_0 is linear or bilinear, so is ϕ_0.

The main difference between (3.16) and (3.8) is that the term with θ_ψ in (3.16) does not have a boundary layer in K since only the interior part of θ_ψ (ref. Fig. 3.1) is used in computing ϕ; whereas θ of (3.8) usually has a boundary layer in K. The last term in (3.16) is new. Since it is a linear combination of ϕ_0, it is smooth in K and does not cause any additional problem. Therefore, using (3.13), (3.16), and summation by parts, we obtain an improved rate of convergence, $O(h^2 + \epsilon|\log(h)|)$, for $(u^h - u)$ in the L^2 norm. It should be mentioned that the base functions constructed from the sampling functions may be discontinuous at the element boundaries. In general, there may exist an $O(\epsilon)$ jump in the base functions across ∂K. Thus, the elements are weakly non-conforming. This makes the analysis of the over-sampling method a little more involved technically. We will report detailed analysis of the over-sampling in the context of multiple scale problems in a subsequent paper [16]. On the other hand, our numerical tests show that the multiscale method with the over-sampling technique indeed works very well.

For problems with continuous scales, which are the main interest of this paper, we note that different scales generate boundary layers with different thicknesses in the sampling domain S. Thus, to avoid the resonant sampling at the grid scale, H should be a couple of times larger than h. At the first sight, this is computationally not attractive since there is too much redundant work. However, we can avoid this difficulty by dividing the computational domain into several large sampling regions. Each sampling region can be used to compute many base functions for the elements contained inside the region (see §4.4).

4. NUMERICAL IMPLEMENTATION AND TESTS

4.1. Implementation

The multiscale method given in §2 is fairly straightforward to implement. Here, we outline the implementation and define some notations that are used frequently in the discussion below. The over-sampling scheme presented in §3.4 will be studied in §4.4. We consider solving problems in a unit square domain. Let N be the number of elements in the x and y directions. The mesh size is thus $h = 1/N$. To compute the base functions, each element is discretized into $M \times M$ subcell elements with mesh size $h_s = h/M$.

In most cases, we use the linear elements to solve the subcell problem for the base functions. If the coefficients a is differentiable and h_s resolves the smallest scale in a, then ϕ^i are computed with second order accuracy. The volume integrals

$$\int_K \nabla\phi^i \cdot a \cdot \nabla\phi^j \, d\mathbf{x} \qquad \text{and} \qquad \int_K \phi^i f \, d\mathbf{x},$$

which are entries of the local stiffness matrix and the right hand side vector, are computed using the two-dimensional centered trapezoidal rule. The results are second order accurate. The amount of computation in the first integral can be reduced by recasting the volume integral into a boundary integral using (2.4). However, we found that this approach may yield a global stiffness matrix that is not positive definite when the subcell resolution is not sufficiently high.

We use a multigrid method with matrix dependent prolongation [27] to solve both the base functions and the large scale problems. We also use this multigrid method and the linear finite element method to solve for a well-resolved solution. This version of the multigrid method has been found to be very robust for 2-D second order elliptic equations (for details, see [27]). Our numerical tests indicate that the number of multigrid iterations is almost independent of the small scales of the problem (see §4.5).

The algorithms are implemented in double precision on an Intel Paragon parallel computer with 512 processors, using the MPI message passing library provided by Intel. Concurrency is achieved through pure data distribution. No special effort is made to improve the parallel efficiency: at the coarse grid level, processors are left idle if no coarse grid data are distributed to them. Only one communication operation, a boundary exchange, is needed for the restriction and prolongation operators in the multigrid iterations. To facilitate the implementation of the multigrid solver of [27] on a multi-computer, the original smoothing method, incomplete line LU decomposition (ILLU), is replaced by a four-color Gauss-Seidel iteration (GS). This requires four boundary exchanges per iteration. If point Jacobi smoothing is used, only one boundary exchange is needed. However, it was found to be very inefficient and required longer CPU times. We find that the number of multigrid iterations using GS can be 1.5 to 2 times larger than that of using ILLU, but the difference in the CPU time is less significant since the GS iterations are cheaper. For convenience, denote these two versions of multigrid as MG-ILLU and MG-GS. In the multiscale method, we can use either one of them to solve the subcell problems, as long as the solutions are computed on a single processor. The parallel MG-GS is used whenever the solutions of the linear systems are computed using more than one processor.

4.2. Cost of the Method

The applicability of an algorithm, in practice, is always limited by the available computer memory and CPU time. For multiple scale problems, these concerns are often crucial. Here, we discuss the cost of the multiscale finite element method (MFEM) in these two aspects. In these regards, it is useful to compare our method with other existing numerical algorithms.

To make the comparison, we consider three popular methods: the conventional finite element method with linear base functions (LFEM), the method base on multiple-scale expansions and cell problems (e.g., [14]), and the methods of local numerical upscaling (e.g.,[12]). Furthermore, for the last two methods, we assume that LFEM is used to solve the cell (or grid block) problems and the effective equation on the coarse grid.

First, we notice that MFEM and the local upscaling methods (e.g. [12]) are similar in terms of memory requirement and operation counts. In fact, the fine scale problems defined on the grid blocks in the local upscaling methods are computationally equivalent to the subcell problems for the base functions in MFEM. For a rectangular mesh, MFEM is a little more expensive since three base functions need to be solved in each element (the fourth one can be computed from (2.8)). In comparison, the local upscaling methods only require solving two fine scale problems to obtain the effective conductivity tensor. The costs of the two methods are the same if triangular elements (grid blocks) are used. However, we note that the local upscaling methods are difficult to implement for triangular grid blocks due to the difficulty in specifying the boundary condition for the fine scale problems (ref. §1). In this regard, MFEM has more flexibility to model complicated geometries. In the future, we plan to perform an extensive numerical study to compare accuracy and efficiency of these two approaches.

Next, we compare MFEM with LFEM and the method based on the multiple scale expansion. Let the number of elements and the number of subcell elements in each dimension be N and M, respectively. The total number of elements at the subcell level is $(NM)^n$, where n is the dimension. Therefore, for LFEM using the same fine grid at the subcell level, the size of the discrete problem and the memory needed is $O(N^nM^n)$. If MFEM is implemented on a serial computer, the corresponding estimate is $O(N^n + M^n)$. The saving of memory implies that MFEM can solve much larger problems than LFEM. To be more specific, on a Sun Sparc20 workstation, our double precision LFEM program takes about 48MB of memory for solving a problem with

$N = 512$. With 12% more memory, total of 54MB, we can solve the problem with $N = 512$ and $M = 128$ using MFEM. Thus the effective resolution increases by a factor of 100. This, however, is an extreme case. In practice, one would like to use large N but relatively small M to include more small scales in the final solution, e.g., $M = 32$ as in many of our numerical tests. Even so, the LFEM program still requires about 49GB of memory to achieve the similar resolution of MFEM. This comparison shows that the multiscale method is well adapted to work station class of computers with limited memory.

On a multi-computer, such as the Intel Paragon, with P processors, the memory required on each processor by LFEM is $O((NM)^n/P)$. For MFEM, if the subcell problems are solved on a single processor, which provides the maximum efficiency, the memory used on each processor is $O(N^n/P + M^n)$. Thus, for $M^n < N^n/P$, which is usually the case in practice, we have a factor of $O(M^n)$ saving in the memory, similar to that in the sequential case. Given a maximum N^n degrees of freedom which can be handled by LFEM, MFEM can always handle M^n times more, where M is only limited by the memory available on each processor but is independent of P. For example, using 256 processors with 32MB memory on each processor, our 2-D parallel LFEM program can solve a problem using 4096^2 elements; again, taking $M = 32$, MFEM can easily deal with 1000 times more elements, which is impossible for LFEM. The difference is even greater in 3-D. It should be noted that other implementations are also possible, e.g., we may solve the subcell problems on several different subsets of processors, so that the limitation on M can be practically removed. This can be done without much effort in MPI as it provides functions of managing groups and communicators.

The memory saving of MFEM comes at the price of more computations. For the same fine grid resolution, if the multigrid method is used, the operation count is $O(N^n M^n)$ for LFEM and $O(N^n + (d-1)N^n M^n)$ for the multiscale method, where d is the number of nodal points on each element. Thus, the ratio of the operation counts in MFEM and LFEM is about $d - 1$. Therefore, triangular and tetrahedra elements are most efficient to use for MFEM in two and three dimensions, where $d-1 = 2$ and 3 respectively. Moreover, the ratio of operation counts is a conservative estimate for the ratio of CPU times on parallel computers since the communication costs of the two methods are different (see §4.3). Note also that, this comparison is made for solving just one particular problem. It is common in practice that multiple runs are desirable for the same medium but with different boundary conditions or source terms. In this case, only $O(N^n)$ operations are needed by MFEM in the later runs since the small scale information, stored in the stiffness and mass matrices, needs not be computed again.

The method based on multiple scale expansions serves the same purpose as MFEM and the local upscaling methods. As we mentioned before, the multiple scale expansions cannot treat problems without scale separation. Here we note that even for problems with scale separations, the method based on multiple scale expansions could be much more expensive than MFEM and the local upscaling methods. For example, suppose there are n_s separable scales characterized by $\mathbf{x}/\epsilon_j$ ($j = 1, \ldots, s$) in a problem. By introducing additional n_s new fast variables, $\mathbf{y}_j = \mathbf{x}/\epsilon_j$, one can derive an effective equation using the multiple scale expansions. Then the total dimension of the cell problems becomes $n_s n$, and hence the operation count is $O(N^n + (M^{n_s} N)^n)$, which increases exponentially as the number of scale increases. Therefore, the method is not practical for problems with multiple separable scales, although it gives accurate effective solutions for special problems with $n_s = 1$ and periodic coefficients.

4.3. Convergence of MFEM

Extensive convergence tests for MFEM based on the two-scale model problem have been reported in [21]. Here, we just briefly summarize the results of those tests. The numerical method of obtaining "exact" solutions for the test problems is also explained. The application of MFEM to composite material and porous flow simulations is given in §5. To facilitate the comparison among different schemes, we use the following shorthands: MFEM-L and MFEM-O indicate that LFEM is used to solve the base functions with linear and oscillatory boundary conditions (see §2.2), respectively.

Because it is very difficult to construct a genuine 2-D multiple scale problem with an exact solution, resolved numerical solutions are used as the exact solutions for the test problems. In all numerical examples below, the resolved solutions are obtained using LFEM. We solve the problems twice on two meshes. Both meshes resolve the smallest scale ϵ and one mesh size is twice as large as the other. Then the Richardson extrapolation is used to compute the "exact" solutions from the solutions on the two meshes. During the tests, we keep the coarser mesh size to be less than $\epsilon/10$, so that the error in the extrapolated solution is less than 10^{-7}. All computations are performed on a unit square, $\Omega = (0,1) \times (0,1)$.

In [21], we confirm the $O(\epsilon/h)$ estimate given in §3.2 (see also below). According to our tests, the numerical error is still small even with $\epsilon/h = 0.64$. This suggests that the error constants are small. By using the spectral method to solve the subcell problems we are able to obtain very accurate base functions. We find that the accuracy of the base functions does not have significant influence on the solution U^h. Computing ϕ^i, A^h, and f^h to second order accuracy seems to be good enough. The boundary layer structure of the first order corrector of the base function is confirmed by our numerical computations (see also §4.4). In addition, we illustrate that the boundary layers can sometimes be removed by using the oscillatory boundary condition given in §2.2, which results in significant improvement in the accuracy of MFEM. In our tests, the oscillatory boundary condition often gives more accurate results than the linear boundary condition because the boundary layer of θ^i using the oscillatory boundary condition is weaker than that using the linear boundary condition. We also provide an example to show that the removal of the boundary layers is sufficient but not necessary for improving the convergence rate.

4.4. Improved Convergence with Over-Sampling

As discussed in §3.4, the over-sampling strategy can be used to remove the resonance effect. The direct implementation of over-sampling, as depicted in Fig. 3.1, is not very efficient due to the redundancy of computation, especially when h is close to ϵ. In the numerical tests below, we decompose the domain into a number of large sampling regions. Each of these sampling regions contains many computational elements. The majority of the computational elements are in the interior of a sampling region. In this simple implementation, there are no redundant computations. In fact, there is a slight reduction in the CPU time (see §4.6). On the other hand, this approach does not guarantee that all the correctors for the base functions are free of boundary layers. Those base functions next to the boundary of the sampling regions are still influenced by the boundary layers in θ_ψ. However, since $H \gg h$ in practice, the boundary layers occupy much smaller regions. Thus, the boundary layer effect is much weaker than that in the original MFEM. From our numerical experiments for problems with and without scale separation, this strategy seems to produce nearly optimum results predicted by our analysis, i.e. $O(h^2 + \epsilon|\log(h)|)$ convergence in L^2 norm.

TABLE 4.1
Results for $\epsilon = 0.005$.

Mesh		MFEM-O			MFEM-L		
N	M	$\|E\|_\infty$	$\|E\|_{l^2}$	rate	$\|E\|_\infty$	$\|E\|_{l^2}$	rate
32	64	4.89e-5	2.52e-5		1.79e-4	9.73e-5	
64	32	1.06e-4	5.79e-5	-1.20	3.86e-4	2.13e-4	-1.13
128	16	1.74e-4	9.65e-5	-0.74	7.32e-4	4.10e-4	-0.94
256	8	3.76e-4	2.10e-4	-1.12	1.40e-3	7.83e-4	-0.93
512	4	1.77e-4	9.88e-5	1.09	1.00e-3	5.61e-4	0.48

TABLE 4.2
Results for $\epsilon/h = 0.64$ and $M = 16$.

N	ϵ	MFEM-O		MFEM-L		LFEM	
		$\|E\|_{l^2}$	rate	$\|E\|_{l^2}$	rate	MN	$\|E\|_{l^2}$
16	0.04	6.23e-5		3.54e-4		256	1.34e-4
32	0.02	8.43e-5	-0.44	3.90e-4	-0.14	512	1.34e-4
64	0.01	9.32e-5	-0.14	4.04e-4	-0.05	1024	1.34e-4
128	0.005	9.65e-5	-0.05	4.10e-4	-0.02	2048	1.34e-4

In the following example, we test the over-sampling scheme by solving (2.1) with

$$\begin{aligned} a(\mathbf{x}/\epsilon) &= \frac{2 + P\sin(2\pi x/\epsilon)}{2 + P\cos(2\pi y/\epsilon)} + \frac{2 + \sin(2\pi y/\epsilon)}{2 + P\sin(2\pi x/\epsilon)}, \\ f(\mathbf{x}) &= -1 \qquad \text{and} \qquad u|_{\partial\Omega} = 0, \end{aligned} \tag{4.1}$$

where $P = 1.8$. The computation is done on a uniform rectangular mesh with N and M being the numbers of elements and subcell elements in each direction, respectively. Note that the analysis of the resonance effect is carried out for triangular elements. Here, we use rectangular elements because the multigrid solver we use is designed for rectangular meshes. In fact, due to our choice of the coefficient a in (4.1), the effective conductivity is a constant diagonal matrix. In this case, one can verify that our analysis is still valid.

The results of MFEM-O, MFEM-L, and LFEM are shown in Tables 4.1 and 4.2. In the tables $E = U - U^h$ is the discrete error at nodal points. Table 4.1 indicates that the errors of MFEM-O and MFEM-L are proportional to h^{-1}. Combining the results of Table 4.1 and Table 4.2, we conclude that the errors of both MFEM-O and MFEM-L are proportional to $O(\epsilon/h)$. We also note that the error of MFEM-O is several times smaller than that of MFEM-L. This is because the oscillatory boundary condition produces a weaker boundary layer in θ^i than the linear boundary condition does, see Fig. 4.1. The procedure of computing θ^i can be found in [21]. Clearly, the structure of θ^i agrees with our theoretical analysis in §3.3.

Let $M_S = H/h_s$, which is the size of the over-sampling problems. For a given fine mesh (i.e., h_s) M_S determines H. We repeat the computations in Tables 4.1 and 4.2 using the over-sampling method with $M_S = 128$ and 256. We use the oscillatory μ^i (see §2.2) as the boundary conditions for the temporary base functions ψ^i. The results are shown in Tables 4.3 and 4.4. Compared with Table 4.1 and 4.2, we can clearly see the improvement in convergence. In Table 4.3, for fixed ϵ the

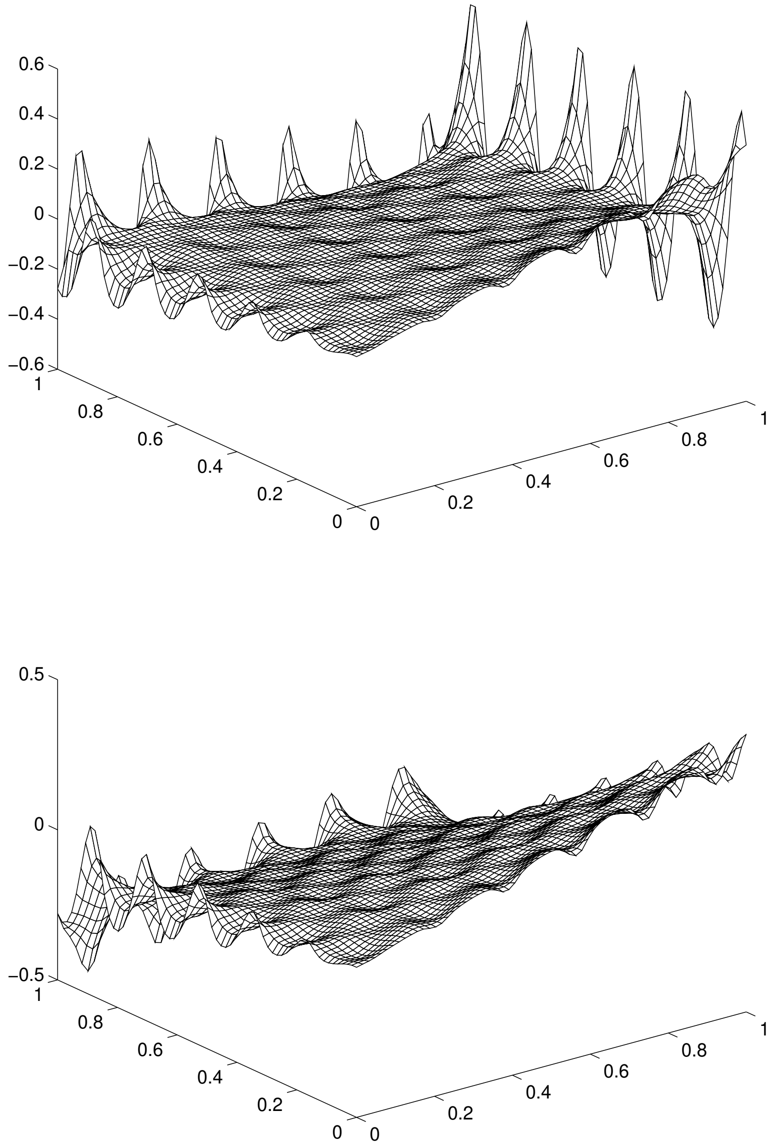

FIG. 4.1. *Surface plots of the first order correctors of the base functions with linear (top) and oscillatory (bottom) boundary conditions* ($\epsilon/h = 0.08\sqrt{5}$).

error remains about the same as h decreases. This is in contrast to the computations presented in Table 4.1 where the errors increase monotonically as h decreases. Moreover, in Table 4.4, the solution converges for fixed ϵ/h as ϵ decreases. We see that the convergence for the $M_S = 256$ case in Table 4.4 is very close to $O(\epsilon)$. On the other hand, the $M_S = 128$ case is not as good due to stronger boundary layer effect (see below). Figure 4.2 shows the first order corrector of the base function constructed using the over-sampling technique. The element in the figure is away from the boundary of the sampling region, and thus there is no boundary layer.

To further understand the results, we recall from the analysis of [21] that the boundary layers of θ^i in each element contribute an $O(\sqrt{\epsilon h})$ error in the H^1 norm. Therefore, the total contribution

TABLE 4.3
Reulsts for the over-sampling method ($\epsilon = 0.005$).

Mesh		$M_S = 128$		$M_S = 256$	
N	M	$\|E\|_\infty$	$\|E\|_{l^2}$	$\|E\|_\infty$	$\|E\|_{l^2}$
32	64	3.08e-5	1.53e-5	3.59e-5	8.14e-6
64	32	4.99e-5	2.06e-5	3.32e-5	1.14e-5
128	16	4.65e-5	1.51e-5	4.42e-5	8.07e-6
256	8	3.66e-5	1.63e-5	2.53e-5	7.26e-6
512	4	1.64e-5	3.42e-6	1.63e-5	5.04e-6

TABLE 4.4
Results for the over-sampling method ($\epsilon/h = 0.64$, $M = 16$).

N	ϵ	$M_S = 128$			$M_S = 256$		
		$\|E\|_\infty$	$\|E\|_{l^2}$	rate	$\|E\|_\infty$	$\|E\|_{l^2}$	rate
16	0.04	3.12e-4	5.78e-5		1.61e-4	5.49e-5	
32	0.02	1.56e-4	2.97e-5	0.96	1.55e-4	2.96e-5	0.89
64	0.01	8.83e-5	1.85e-5	0.68	8.16e-5	1.54e-5	0.94
128	0.005	4.65e-5	1.51e-5	0.29	4.42e-5	8.07e-6	0.93

due to the boundary layers in all elements is $O(\sqrt{\epsilon/h})$ (since the number of elements is proportional to h^{-2}). This is basically how the leading order term in (3.7) is obtained. Roughly speaking, in the present implementation of the over-sampling technique, there are $O(1/(hH))$ elements which contain the boundary layers of θ_ψ. Therefore, the total H^1-norm error due to the boundary layers is $O(\sqrt{\epsilon/H})$. On the other hand, from the discrete error analysis of §3.3, we can estimate the l^2-norm error being roughly $O(\epsilon/H)$. Since $H = M_S h_s$, these estimates explain why the solutions are more accurate for larger M_S in most of the tests with fixed h_s in Table 4.3. We have repeated the computation in Tables 4.3 and 4.4 using a single sampling domain $S = \Omega$ with $H = 1$, and we observed an $O(\epsilon)$ convergence (not shown here). It should be noted that the numerical results of the over-sampling technique in Tables 4.3 and 4.4 are better than the $O(\epsilon/H)$ estimate. In fact, in Table 4.4, $\epsilon/H \approx 0.1$ is fixed. According to the above estimate, the solutions should not converge. This discrepancy may be due to the small error constants in the leading order estimates. We will study this issue in more details in our coming paper [16].

We also find that changing the boundary condition for ψ^i to linear functions have no significant effect on the convergence, especially when H is large. However, since the boundary layer is stronger, the solution is less accurate. The degradation is smaller for larger H. Another interesting phenomenon is that the solutions using MFEM with the over-sampling technique can be more accurate than the resolved direct solutions using LFEM on a fine mesh h_s. Intuitively, one would think that the resolution of the direct solution on a fine grid h_s should be higher than that of the MFEM on a coarser grid h.

We stress that the present implementation of the over-sampling scheme is simple but not ideal. A modification is to enlarge the size of those sampling domains away from ∂S by $O(\epsilon)$. This will completely remove the boundary layer effect due to the interior boundaries of the sampling regions while the amount of redundant work is kept small.

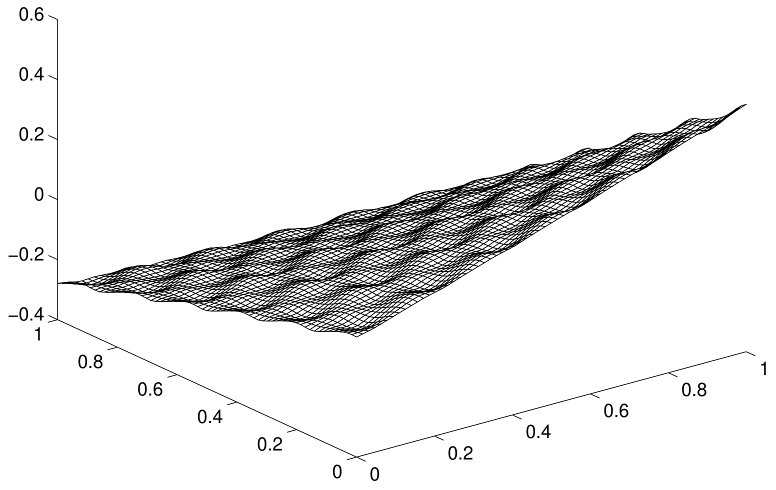

FIG. 4.2. *First order corrector of the base function, which is is constructed from over sampling* ($\epsilon/h = 0.08\sqrt{5}$).

4.5. Multigrid Convergence

As we mentioned before, we solve the discrete linear system resulting from our multiscale FEM by a multigrid solver that uses a matrix dependent prolongation operator. It has been observed in the multigrid literature that the number of multigrid iterations usually deteriorates significantly for elliptic problems with rough coefficients and/or highly oscillating coefficients, see e.g., [11, 18]. This would slow down the speed of the overall solution procedure. Therefore, it is important to design a multigrid method for which the number of multigrid iterations is essentially independent of the mesh size and the small scale features in the solution. Another difficulty for multigrid methods comes from the high contrast in the coefficient a, defined as $C_a = \max(a)/\min(a)$. In practice C_a can be very high: an order of 10^7 to 10^8 is typical in groundwater applications. Thus it is equally important that the convergence in the multigrid iterations should be insensitive to the contrast in the coefficient.

Our numerical experiments show that the multigrid method given in [27] applied to a traditional FEM is rather robust when the problem is well-resolved on the fine grid. This is a nontrivial accomplishment, because a standard multigrid method would give a much poorer convergence rate. The success lies in the matrix dependent prolongation, which passes important fine grid information onto the coarse grid operators. However, when the problem is under-resolved in the fine grid, even the multigrid method with a matrix dependent prolongation gives a very poor convergence rate.

In our MFEM formulation, the problem is directly discretized on a relatively coarse grid, whose mesh size is typically larger than the smallest scale in the solution. The discrete solution operator is constructed using the multiscale base functions. Our numerical experiments show that the multigrid convergence for the resulting discrete linear systems is independent of ϵ and h. For example, it typically takes the parallel MG-GS solver 12 or 13 iterations to compute the MFEM solutions of (2.1) given in §4.3. The number of iterations is independent of ϵ and h in the calculations presented in Tables 4.1 and 4.2.

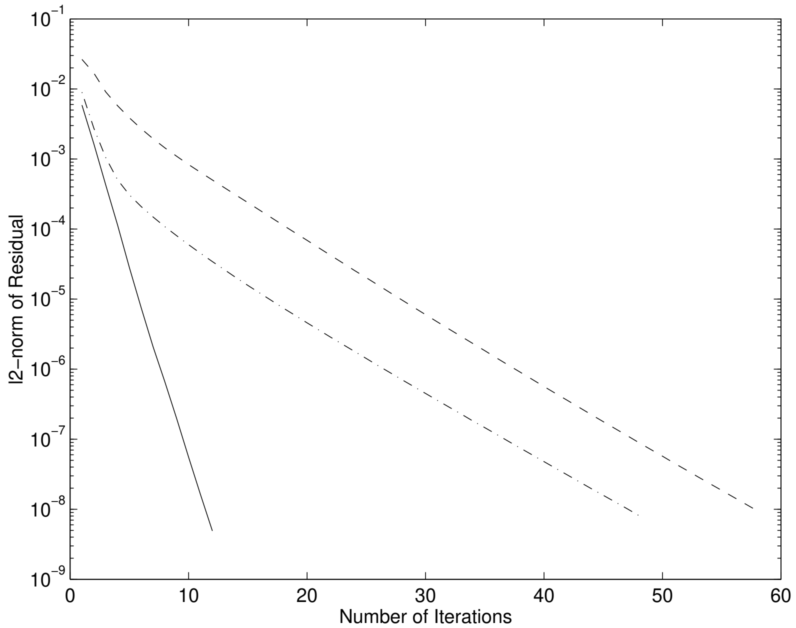

FIG. 4.3. *Convergence of multigrid iteration for solving* (2.1) *and* (4.2) *with* $C_a = 1.6 \times 10^5$ *and* $\epsilon = \sqrt{2}/1000$. *Solid line: MFEM* ($N = 256,\ M = 32$)*; dash line: LFEM* ($N = 256$)*; dashdot line: LFEM* ($N = 512$).

To test how the multigrid convergence depends on C_a, we solve (2.1) with

$$\begin{aligned} a(\mathbf{x}) &= \frac{1}{(2 + P\sin(2\pi x/\epsilon))(2 + P\sin(2\pi y/\epsilon))}, \\ f(\mathbf{x}) &= -1, \qquad \text{and} \qquad u|_{\partial\Omega} = 0, \end{aligned} \tag{4.2}$$

where P controls the contrast C_a. In this test, we choose $\epsilon = \sqrt{2}/1000$ and solve the problem using MFEM with $N = 256$ ($M = 32$), and LFEM with $N = 256$ and $N = 512$. Note that with $\epsilon = \sqrt{2}/1000$, $N = 256$ or $N = 512$, the problem is under-resolved in the LFEM calculations. The parallel MG-GS solver is used to solve the discrete systems of equations. The multigrid convergence for $C_a = 1.6 \times 10^5$ is given in Fig. 4.3. We see that it takes significantly more iterations for MG-GS to converge in the LFEM calculations than in the MFEM calculation. We also plot the dependence of the multigrid convergence on the contrast coefficient, C_a, in Fig. 4.4. We can see that the multigrid convergence for LFEM depends strongly on C_a; whereas the multigrid convergence for MFEM is basically independent of C_a. The reason for the poor multigrid convergence in the LFEM calculations is due to the fact that LFEM does not sample the correct small scale information in the fine grid. In comparison, MFEM captures correctly the small scale information in its finest level of grid, h, which is still larger than the smallest scale, ϵ, in the solution. These numerical experiments demonstrate that the multiscale base functions are also valuable for obtaining optimum multigrid convergence using a relatively coarse grid to compute highly heterogeneous, multiscale problems.

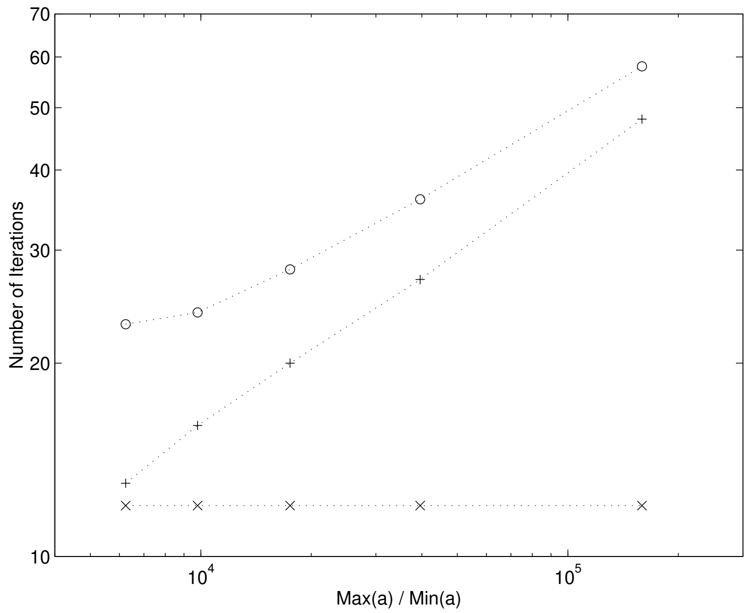

FIG. 4.4. *The dependency of multigrid convergence on C_a for solving* (2.1) *and* (4.2) *with* $\epsilon = \sqrt{2}/1000$. $\times$*: MFEM* ($N = 256$, $M = 32$)*;* $\circ$*: LFEM* ($N = 256$)*;* $+$*: LFEM* ($N = 512$)*.*

4.6. Parallel Performance

In this subsection, we provide some speedup timing results of MFEM and compare them with those of LFEM. The results are shown in the logarithmic execution-time plots, which plot the execution times against the number of processors used. The test problem in §4.3 is solved on a fine grid with $MN = 1024$ elements in x and y directions using an increasing number of processors. For MFEM, we solve the problem with $M = 16$ and 32, which are represented in Figs. 4.5 to 4.8 by $\times$ and $+$, respectively. The LFEM solution using the parallel MG-GS multigrid solver is denoted by $\circ$. The dotted straight lines represent the ideal linear speedup. For all multigrid iterations, the tolerance is set to 1×10^{-8}.

The results for the total CPU time (excluding the time for input and output) of solving the problem by using LFEM and MFEM are shown in Figs. 4.5 and 4.6. Fig. 4.5 shows the CPU times of using MFEM with MG-ILLU for solving the base functions and the parallel MG-GS for solving the large scale solutions. The CPU time of using LFEM is also shown in the figure for comparison. We see that the speedup of MFEM follows very closely the linear speedup, while that of LFEM does not. For both methods, the departure from the linear speedup is mainly due to the communication at the coarse grid levels. However, for MFEM, this occurs only when the large scale solution is computed. In another implementation, we gather the data onto a single processor and solve the large scale problem on that processor. For small N, hence large M (NM is fixed), such an approach is more efficient than the previous one. The improvement in the speedup is shown in Fig. 4.6. When N is large, multiple processors should be used to solve the large scale problem.

These figures also indicate that for MFEM the computation is more efficient with larger subcell problems. Therefore, for both efficiency and accuracy reasons, it is desirable to choose the size of

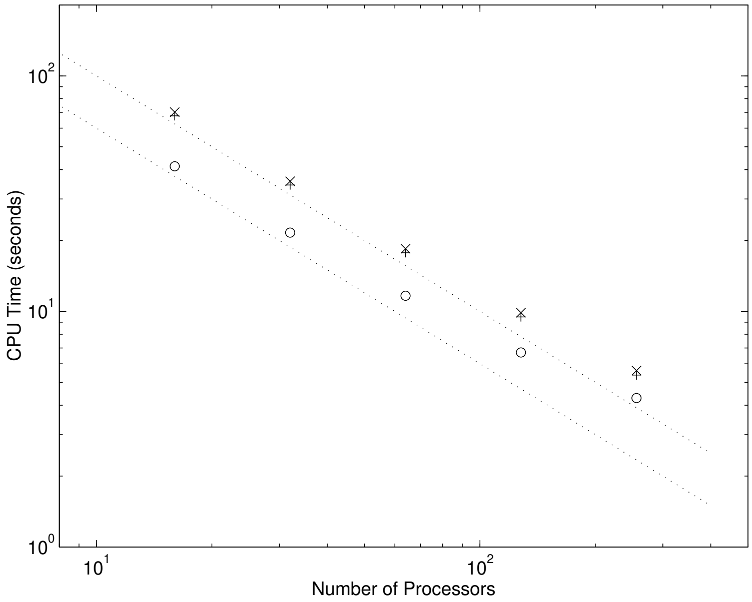

FIG. 4.5. *Total CPU time used by LFEM ($\circ$) and MFEM-O with MG-ILLU for computing the subcell solutions. $\times$: $M = 16$, $+$: $M = 32$.*

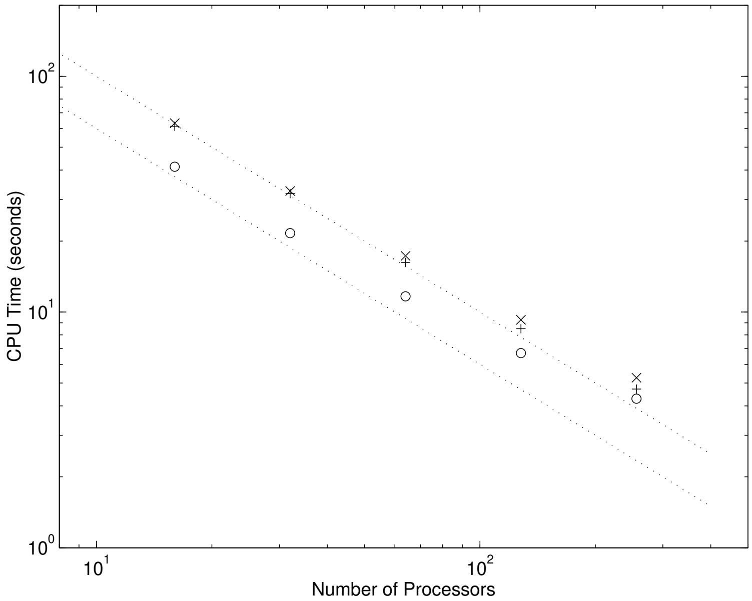

FIG. 4.6. *The same as Fig. 4.5, except that for MFEM the large scale solution is obtained on a single node.*

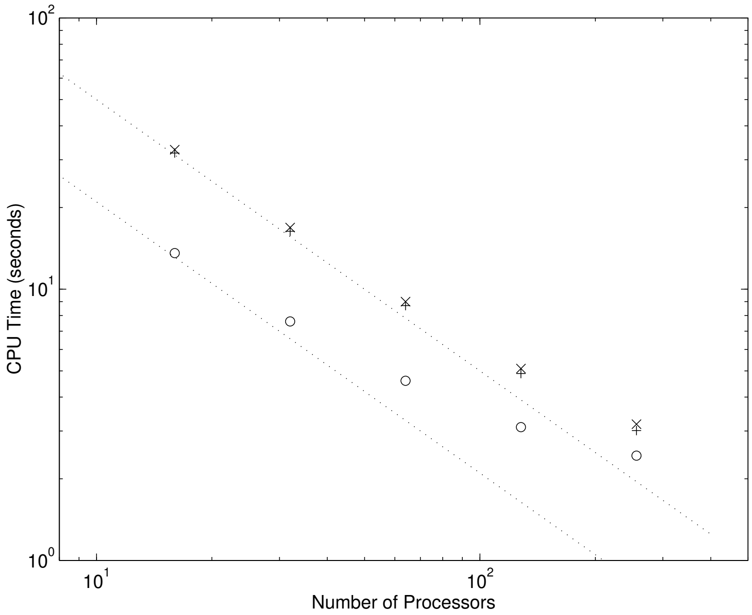

FIG. 4.7. *A comparison of CPU time used by multigrid iterations in the LFEM (○) and MFEM computations. For the latter, it includes the time for solving base functions and the large scale solution.* $\times$*:* $M = 16$*,* $+$*:* $M = 32$*.*

sampling domain (i.e., M_S) as large as possible. On the other hand, given M_S, the choice of M has no significant effect on the CPU time. We also note from Figs. 4.5 and 4.6 that the time used by the multiscale method is only about 50% more than that used by LFEM if run on 16 processors; moreover, the percentage drops down quickly (as low as 9% for 256 processors, see Fig. 4.6) as the number of processors increases. In contrast, the difference is about 95% for sequential runs. This can be partially attributed to the better parallel speedup of MFEM. More importantly, as mentioned before, MG-ILLU converges faster than MG-GS. The flexibility of using various fast sequential linear solvers for the subcell problems is very useful in practice.

Note that a significant amount of the total CPU time is used to setup the linear system of equations in the LFEM computation. Similarly, in the MFEM computation, discrete linear systems are computed for both the base functions and the large scale solution. Therefore, the comparisons in Figs. 4.5 and 4.6 do not reflect the operation counts given in §4.2. In Figs. 4.7 and 4.8, the CPU times for multigrid iterations alone are compared. For MFEM, this includes the multigrid iterations for solving the base functions and the large scale solution. The trends shown in Figs. 4.7 and 4.8 are similar to those in Figs. 4.5 and 4.6: MFEM spends 130% more time than LFEM on 16 processors and 13% (Fig. 4.7) or even -8% (Fig. 4.6) more time on 256 processors.

It should be noted that it is quite difficult to make a "fair" comparison between the CPU times of MFEM and LFEM due to many factors. In fact, such a comparison may not be very meaningful since the goals of the two methods are so different. Our goal for MFEM is to provide a method that can capture much more small scale information than a direct method can resolve. Our experiments illustrate that we can achieve this goal with a small amount of extra work. Furthermore, the speedup comparisons do indicate that MFEM adapts very well to the parallel computing environment.

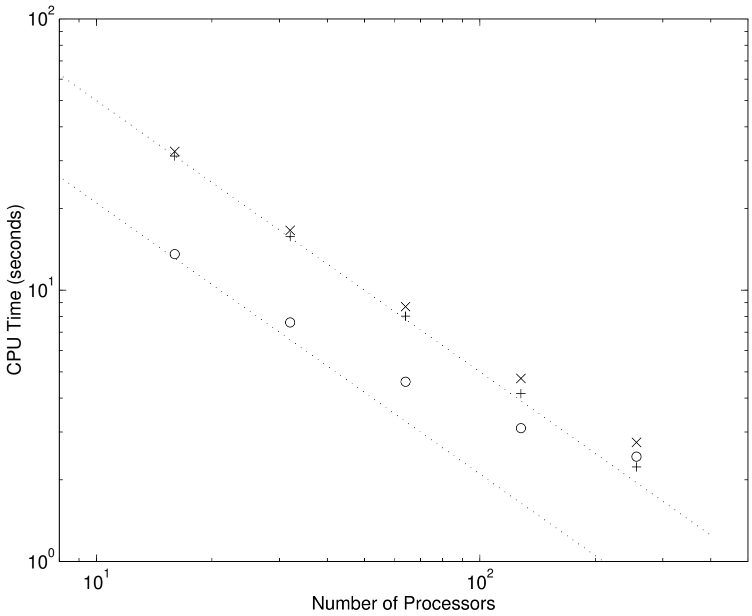

FIG. 4.8. *The same as Fig. 4.5, except that for MFEM the large scale solution is obtained on a single node.*

5. APPLICATIONS

In this section, we apply the multiscale method to problems with continuous scales, including steady conduction through fiber composites (§5.1) and steady flows through random porous media (§5.2). The problems we solve are models of the real systems. Both types of problems are described by (2.1). The conductivity of the composite materials and the permeability of the porous media are represented by the coefficient $a(\mathbf{x})$. In reality, the properties of composite materials and porous media may undergo abrupt changes, which correspond to jump discontinuities in $a(\mathbf{x})$. Such discontinuities should be treated with special care in order to get accurate solutions. Here, to simplify the numerical experiments, we will not consider the abrupt changes. We, however, allow the conductivity or permeability to vary rapidly and continuously.

5.1. Unidirectional Composites

Consider steady heat conduction through a composite material with tubular fiber reinforcement in a matrix (see Fig. 5.1). The problem is described by (2.1) with the coefficient $a(\mathbf{x})$ representing the conductivity of the material. This is referred to as a unidirectional composite in [4], for the local conductivity varies rapidly along one direction. Two special finite element methods have been designed in [4] to compute such problems with high accuracy. One of them requires the local alignment of element boundaries with the fibers; the other is more general but it does not allow the coefficient to change abruptly.

Here, we use the multiscale method to solve the problem. Our method is similar to Method III$'$ of [4] in the sense that it does not require the alignment of elements with the fibers. On the other hand, our method is targeted at general 2-D problems with oscillations in both spatial directions.

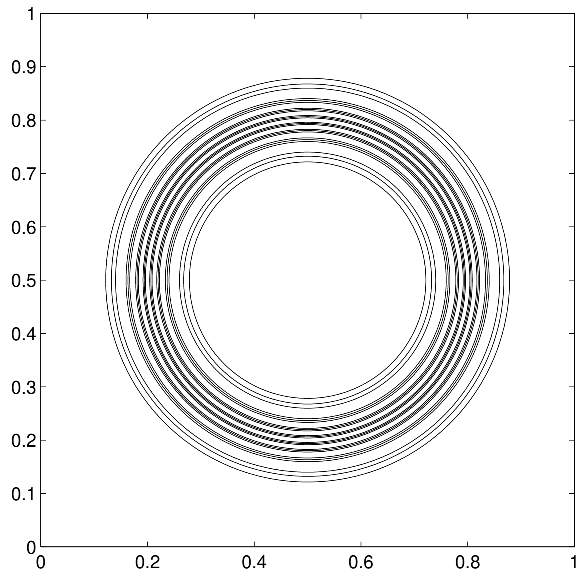

FIG. 5.1. *The model of 2-D unidirectional fiber composite.*

The conductivity of the material is modeled by the following smooth function

$$a(\mathbf{x}) = 2 + P\cos(2\pi \tanh(w(r - 0.3))/\epsilon),$$

where $r = ((x-\frac{1}{2})^2 + (y-\frac{1}{2})^2)^{\frac{1}{2}}$, P controls the ratio between the conductivity of the "fibers" and that of the matrix, w determines the total width of the reinforcement, and ϵ (together with w) sets the wavelength of the local unidirectional oscillation. The structure of $a(\mathbf{x})$ is visualized in Fig. 5.1, where the contour plot of $a(\mathbf{x})$ is given. In the following computation, we take $P = 1.8$, $w = 20$, and $\epsilon = 0.1$. These choices imply that the shortest wavelength in the oscillation is about 0.005, for which we can compute a well-resolved solution for the problem using LFEM and the Richardson extrapolation. The boundary condition is given by

$$u(x, y) = x^2 + y^2 \qquad (x, y) \in \partial\Omega,$$

and a uniform source $f(x, y) = -1$ is specified. We note that the problem has continuous scales.

The problem is solved using MFEM-L, MFEM-O, LFEM, and MFEM with the over-sampling technique. Meshes with different numbers of elements per dimension (N) are used. For all MFEM solutions, M is chosen so that the base functions resolve the smallest scales of the problem; in all cases, $NM = 2048$. Again, we choose $M_S = 256$, which is about the largest number for which the computation of the sampling functions fits in the memory of a single processor on the Intel Paragon computer. The linear and oscillatory boundary conditions for the sampling functions ψ^i are indicated by "os-L" and "os-O", respectively. We note that in this case, the oscillation is localized in the circular region with "fibers". Away from that region, the multiscale base functions are very close to the standard bilinear base functions since the conductivity is practically a constant. On the other hand, the multiscale base functions become oscillatory in the fiber region. In Fig. 5.2, the l^2-norm errors of the solutions are shown. The solid line in the figure represents the line of first order convergence in h; the dash line indicates the solution error of using LFEM on the 2048×2048 fine mesh.

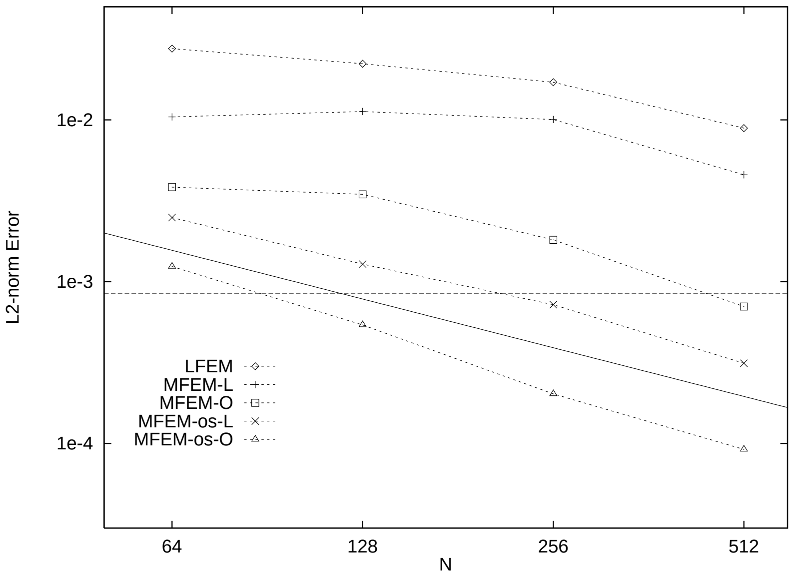

FIG. 5.2. *The l^2-norm error of the solutions using various schemes.*

As in the tests for the two-scale problem in §4.3 and §4.4, Fig. 5.2 shows that the boundary conditions of the base functions have significant influence on the accuracy and the convergence of the solutions: the oscillatory boundary condition is clearly better. By comparing results of MFEM-O and MFEM-os-O as well as MFEM-L and MFEM-os-L, we see a great improvement in the accuracy of solutions using the over-sampling technique. In fact, with either the linear or the oscillatory boundary condition for the sampling functions, the over-sampling technique gives more accurate solutions than both MFEM-O and MFEM-L. Furthermore, the over-sampling technique leads to $O(h)$ convergence, which depends slightly on the boundary conditions for ψ^is. From Fig. 5.2, we observe that the solutions of MFEM with the over-sampling technique become more accurate than the resolved direct solution of LFEM, obtained on the fine mesh, h_s, (compare also Tables 4.2 with 4.4). These results illustrate that MFEM with the over-sampling technique is a good candidate for solving problems of unidirectional fiber composites. In [21], MFEM without the over-sampling technique is also applied to a problem with continuous scales and genuine 2-D oscillations. The results are similar to those reported here. Thus, it is plausible that MFEM is useful for general fiber composite problems. It is worth mentioning that the efficiency of the above computation can be greatly improved by constructing the multiscale functions only in the region of rapid oscillations. Moreover, one may use larger elements in the region with constant conductivity and smaller ones in the region with oscillatory conductivity.

5.2. Flows through Random Porous Media

Computing steady flows through random porous media is very important for studying many transport problems in subsurface formations, such as groundwater and contaminant transport in aquifers. The direct methods (e.g., [1]) and local numerical upscaling methods (ref. [12, 23]) have been applied to this problem. In this subsection, we use the multiscale method and the over-sampling technique to compute steady state single phase flows through random porous media.

5.2.1. Random field generation

To model the random media, we follow the approach in [12]. A random porosity field p is first generated and the permeability field is then calculated from

$$a = \alpha 10^{\beta p},$$

where α and β are scaling constants. If p is normally distributed, then the permeability field has a log-normal distribution, which can represent the areal variation of some real systems [13]. Here, we use the spectral method to generate the Gaussian random distribution for the porosity field. At each point $\mathbf{x}$, the value of p is given by the sum of a number (N_f) of Fourier modes with low to high frequency, which are determined by uniformly distributed random phases in the interval of 0 to 2π. The summation is performed by using the fast Fourier transform (FFT).

One of the advantages of this approach is that we can control the highest frequency N_f of the Fourier modes and hence the smallest scale contained in the porosity field. This control enables us to resolve the permeability field by using a fine mesh. For example, given $N_f = 64$, we may choose $N = 512$ for the fine mesh. Then, there are 9 nodal points per shortest wavelength. Therefore, we may compute accurately resolved solutions for comparison with the MFEM solutions. Another advantage of the spectral method is that the power spectrum of the distribution can be easily manipulated. This provides a convenient way of generating statistically fractal porosity distributions, which are found for many natural porous media [26]. More specifically, the spectral energy distribution of a statistically fractal field has a power-law structure. By constructing random fields with different power-law spectrum, which can be easily done in the Fourier domain, one obtains statistically fractal fields with different fractal dimensions. For a detailed description about the correspondence between the power law and the fractal dimension, we refer to [26]. Because the random porosity fields used in our simulations are very large, they have to be generated on the parallel computer. A parallel FFT is developed for this purpose. In addition, we use a parallel random number generator described in [20] to generate the uniform deviates. A 256×256 image of a random porosity field with the fractal dimension of 2.8 is shown in Fig. 5.3. In the following, we solve (2.1) with $u = 0$ on $\partial\Omega$ and an uniform source $f = -1$. This is a model of flow in an oil reservoir or aquifer with uniform injection in the domain and outflow at the boundaries. As in §5.1, we fix $NM = 2048$ and choose $M_S = 256$.

5.2.2. Results

First, we solve for a log-normal distribution of the permeability with $N_f = 256$; α and β are chosen such that the ratio between the maximum and minimum values of $a(\mathbf{x})$ is 400. We note that the permeability distribution is isotropic. The l^2-norm errors obtained using various schemes are plotted in Fig. 5.4. In this case, the error of using MFEM-L increases initially as h decreases (N increases), which is similar to the results shown in §4.3. This trend reverses when h becomes smaller than the smallest scale of the problem, i.e., $N = 512$. Again, the boundary condition for the base functions makes a big difference in the convergence trend. However, the influence in the accuracy is not as significant. The over-sampling technique clearly improves both the accuracy and convergence. The rate of convergence of MFEM-os-O is lower than that computed in Fig. 5.2, about $O(h^{0.2})$. Nevertheless, such a convergence behavior is important in practice.

Next, in Fig. 5.5 we give the results for the fractal porosity field shown in Fig. 5.3. The parameters of the simulation are the same as above. The fractal dimension 2.8 implies that the

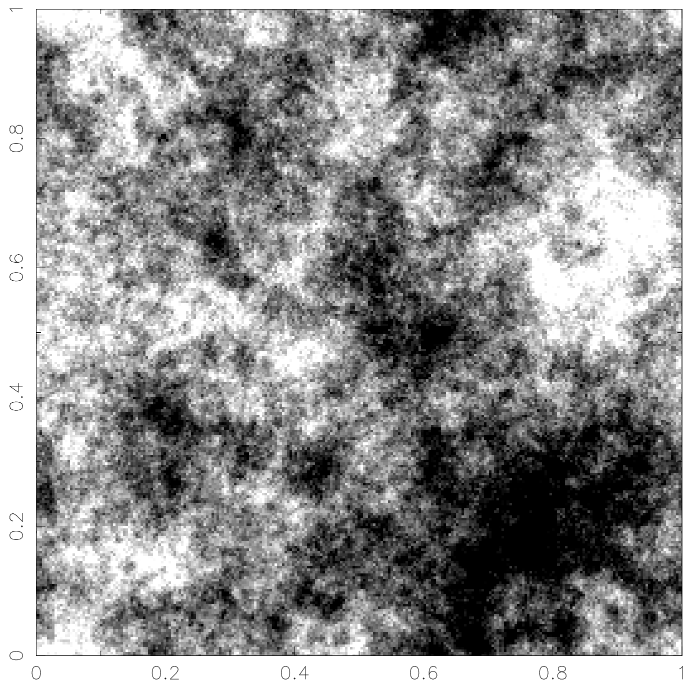

FIG. 5.3. *Porosity field with fractal dimension of* 2.8 *generated using the spectral method.*

spectral energy density decays according to a $(-7/5)$-power law. The decay of the small scales has a positive effect on the accuracy and convergence for all methods. Among them, the over-sampling technique still leads to most accurate results. Note that the convergence rate of MFEM-os-O decreases as N increases. In fact, similar trend is also shown in the previous figure. In both cases, the errors of MFEM-os-O are very close to those of the resolved LFEM solutions (the dash lines). The problem may due to the effect of some residual boundary layers that are not completely removed by the present implementation of the over-sampling technique. We will study this problem in more details in future works. On the other hand, we note that the MFEM with the over-sampling technique is most useful in the unresolved regime where the over-sampling technique performs well. The degeneration in the convergence rate should not be a big concern.

We also note that the relative error of the LFEM solution at $N = 512$ is already less than 0.77%, which is small enough for practical purposes. Thus, due to the decay of small scales, one needs not resolve all the scales in order to get satisfactory solutions. This observation should also be applicable to MFEM. We use MFEM-os-O to compute the problem with $N = 128$ and $M = 4$, which has an equivalent fine grid resolution as LFEM with $N = 512$. The errors of the two solutions are rather close, 7.18×10^{-4} for MFEM-os-O versus 6.86×10^{-4} for LFEM.

In the previous two examples, the permeability fields are isotropic, which can model the areal variations of aquifers. However, the cross section of an aquifer is characterized by the layer structures. Thus, the permeability field is anisotropic. In Fig. 5.6, an image of numerically generated porosity field for a cross section is shown. To generate this field, we let the Fourier modes decay in the x direction according to a given 1-D fractal dimension (1.5 in our case), but the Fourier modes do not decay in the y direction. In this example, we have $N_f = 512$ in the x direction and $N_f = 256$ in the y direction. The resulting distribution along the vertical direction for each

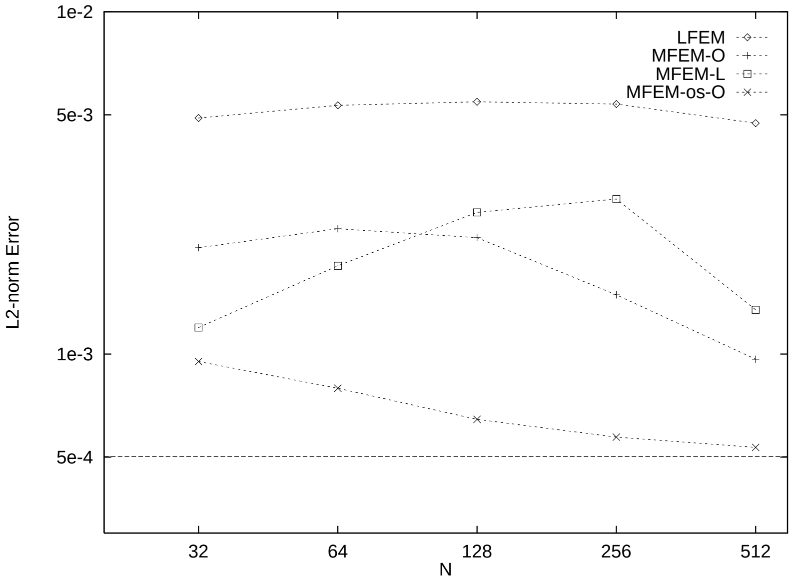

FIG. 5.4. *The l^2-norm error of the solutions using various schemes for a log-normally distributed permeability field. The horizontal dash line indicates the error of the LFEM solution with $N = 2048$.*

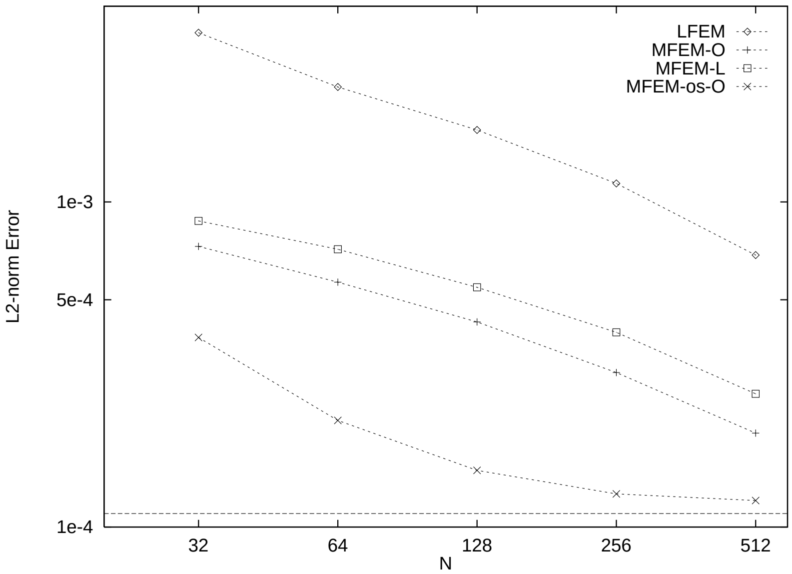

FIG. 5.5. *The l^2-norm error of the solutions using various schemes for a fractally distributed permeability field. The horizontal dash line indicates the error of the LFEM solution with $N = 2048$.*

fixed x is approximately Gaussian. Thus, the permeability varies more rapidly along the vertical direction. For the permeability field, we choose α and β such that the ratio between the maximum and minimum values of a is 10^4.

The numerical errors are plotted in Fig. 5.7. We find that both MFEM-O and MFEM-os-O solutions have about the same accuracy as the resolved LFEM solution on the 2048^2 mesh (the

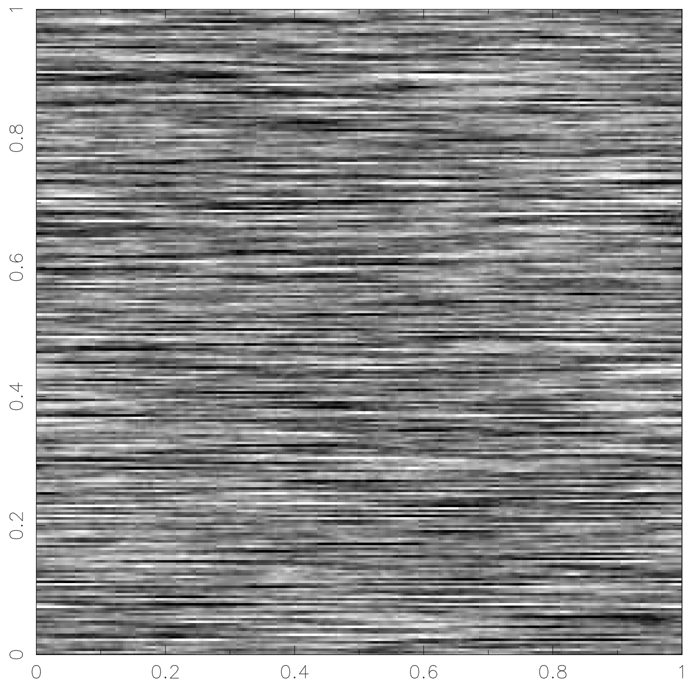

FIG. 5.6. *Porosity field for cross section generated using the spectral method.*

dash line). This is not surprising. We note that the rapid oscillations in the vertical direction align with the mesh. Therefore, the oscillatory boundary condition captures the local property of the differential operator near the element boundaries. This makes the multiscale base functions very effective. For this reason, the over sampling technique does not offer additional improved accuracy over the oscillatory boundary condition. The linear boundary condition, on the other hand, gives a poor convergence result since it cannot "sense" the layer structure. Thus it leads to the resonance effect, as shown in Fig. 5.7.

6. CONCLUDING REMARKS

We have successfully developed a multiscale finite element method for solving elliptic problems in composite materials and porous media. The problems are characterized by the highly heterogeneous and oscillatory coefficients. In our method, the small scale information is captured by the finite element bases constructed from the leading order elliptic operator. In the case of periodic structure, we prove that the method converges to the correct effective solution as $\epsilon \to 0$ independent of ϵ. We have analyzed the "resonant scale" phenomenon associated with upscaling type of methods. To alleviate the difficulty, we propose an over-sampling technique. Our numerical experiments give convincing evidence that the multiscale method is capable of capturing the large scale solution without resolving the small scale details. Applications of the method to practical problems with continuous scales seem promising. We demonstrate that at a reasonable cost, the multiscale method is able to solve very large scale practical problems that are otherwise intractable using the direct methods.

The idea of constructing multiscale base functions is not restricted to the elliptic equations.

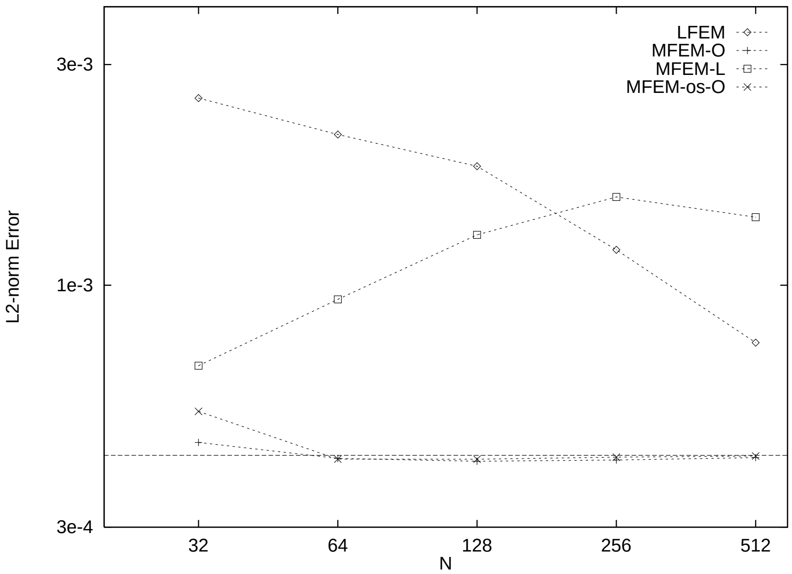

FIG. 5.7. *The l^2-norm error for cross section solutions using various methods. The horizontal dash line indicates the error of the LFEM solution with $N = 2048$.*

In the future, we will apply the multiscale method to solve convection-diffusion equations and the wave equations in multiscale media. Applications such as turbulent transport problems in high Reynolds number flows and wave propagation and scattering in random heterogeneous media will be considered.

Acknowledgments. We thank Professor Bjorn Engquist and Mr. Yalchin Efendiev for many interesting and helpful discussions. This work is supported in part by ONR under the grant N00014-94-0310 and DOE under the grant DE-FG03-89ER25073.

REFERENCES

[1] R. Ababou, D. McLaughlin, and L. W. Gelhar. Numerical simulation of three-dimensional saturated flow in randomly heterogeneous porous media. *Transport in Porous Media*, 4:549–565, 1989.

[2] M. Avellaneda, T. Y. Hou, and G. Papanicolaou. Finite difference approximations for partial differential equations with rapidly oscillating coefficients. *Mathematical Modelling and Numerical Analysis*, 25:693–710, 1991.

[3] M. Avellaneda and F-H. Lin. Homogenization of elliptic problems with L^p boundary data. *Appl. Math. Optim.*, 15:93–107, 1987.

[4] I. Babuška, G. Caloz, and E. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM J. Numer. Anal.*, 31:945–981, 1994.

[5] I. Babuška and E. Osborn. Generalized finite element methods: Their performance and their relation to mixed methods. *SIAM J. Numer. Anal.*, 20:510–536, 1983.

[6] I. Babuška and W. G. Szymczak. An error analysis for the finite element method applied to convection-diffusion problems. *Comput. Methods Appl. Math. Engrg.*, 31:19–42, 1982.

[7] J. Bear. Use of models in decision making. In T. H. Dracos and F. Stauffer, editors, *Transport and Reactive Processes in Aquifers*, pages 3–9. Balkema, Rotterdam, 1994.

[8] A. Bensoussan, J. L. Lion, and G. Papanicolaou. *Asymptotic Analysis for Periodic Structure*, volume 5 of *Studies in Mathematics and Its Applications*. North-Holland Publ., 1978.

[9] D. T. Burr, E. A. Sudicky, and R. L. Naff. Nonreactive and reactive solute transport in 3-dimensional heterogeneous porous media—mean displacement, plume spreading, and uncertainty. *Water Resour. Res.*, 30:791–815, 1994.

[10] M. E. Cruz and A. Petera. A parallel monte-Carlo finite-element procedure for the analysis of multicomponent random media. *Int. j. numer. methods eng.*, 38:1087–1121, 1995.

[11] Jr J. E. Dendy. Black box multigrid. *J. Comput. Phys.*, 48:366–386, 1982.

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

[13] L. J. Durlofsky. Representation of grid block permeability in coarse scale models of randomly heterogeneous porous-media. *Water Resour. Res.*, 28:1791–1800, 1992.

[14] B. B. Dykaar and P. K. Kitanidis. Determination of the effective hydraulic conductivity for heterogeneous porous media using a numerical spectral approach: 1. method. *Water Resour. Res.*, 28:1155–1166, 1992.

[15] W. E and T. Y. Hou. Homogenization and convergence of the vortex method for 2-d euler equations with oscillatory vorticity fields. *Comm. Pure and Appl. Math.*, 43:821–855, 1990.

[16] Y. Efendiev, T. Y. Hou, and X. H. Wu. A multiscale finite element method for problems with highly oscillatory coefficients. 1996. in preparation.

[17] B. Engquist and T. Y. Hou. Particle method approximation of oscillatory solutions to hyperbolic differential equations. *SIAM J. Numer. Anal.*, 26:289–319, 1989.

[18] B. Engquist and E. Luo. Multigrid methods for differential equations with highly oscillatory coefficients. In *Proceedings of the Sixth Copper Mountain Conference on Multigrid Method*, 1993.

[19] J. Frehse and R. Rannacher. Eine l^1-fehlerabschätzung für diskrete Grundlösungen in der Methode der finiten Elemente. In J. Frehse, editor, *Finite Elemente*, number 89 in Bonn. Math. Schrift., pages 92–114. 1975.

[20] B. L. Holian, O. E. Percus, T. T. Warnock, and P. A. Whitlock. Pseudorandom number generator for massively-parallel molecular-dynamics simulations. *Phys. Review E*, 50(2):1607–1615, 1994.

[21] T. Y. Hou, X. H. Wu, and Z. Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. 1996. submitted to Math. Comp.

[22] P. Jussel, F. Stauffer, and T. Dracos. Transport modeling in heterogeneous aquifers. 2. 3-dimensional transport model and stochastic numerical tracer experiments. *Water Resour. Res.*, 30:1819–1831, 1994.

[23] J. F. Mccarthy. Comparison of fast algorithms for estimating large-scale permeabilities of heterogeneous media. *Transport in Porous Media*, 19:123–137, 1995.

[24] R. Scott. Optimal l^∞ estimates for the finite element method on irregular meshes. *Math. Comp.*, 30:681–697, 1976.

[25] A. F. B. Tompson. Numerical-simulation of chemical migration in physically and chemically heterogeneous porous-media. *Water Resour. Res.*, 29:3709–3726, 1993.

[26] D. L. Turcote and J. Huang. Fractal distributions in geology, scale invariance, and deterministic chaos. In C. C. Barton and P. R. La Pointe, editors, *Fractals in the Earth Sciences*, pages 1–40. Plenum Press, 1995.

[27] P. M. De Zeeuw. Matrix-denpendent prologation and restrictions in a blackbox multigrid solver. *J. Comput. Applied Math.*, 33:1–27, 1990.