Multiscale Finite Element Method for Elliptic Problems

Xin Liu Qisi Wang Shu Wang Yaqi Zhang
Yeyu Zhang

May 2017

1 Introduction

1.1 Motivation

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 [1]. For example, the difficulty in analyzing groundwater transport is mainly caused by the heterogeneity of subsurface formulations spanning over many scales. The heterogeneity is often represented by the multiscale fluctuations in the permeability of the media.

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 simulation, 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 [2]. 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.

1.2 Approach

Hou and Wu proposed a multiscale finite element method (MsFEM) for solving partial differential equations with multiscale solutions [1]. The center 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 their 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 [3].

1.3 Outline

The small paper is organized as follows. The formulation of the 2D multiple-scale elliptic problem is reviewed in Section 2.1. The construction of base functions of multiscale finite element method is introduced in Section 2.2. The numerical implementation and comparison with traditional finite element method are studied in Section 3. Using a numerical example, we showed that multiscale finite element outperforms traditional finite element method when solving multiple-scale elliptic problems. Section 4 is reserved for some conclusion remarks.

2 Formulations

MsFEM consists of two major ingredients: multiscale base functions and a global numerical formulation that couples these multiscale base functions [4]. Base functions are designed to capture the multiscale features of the solution. Important multiscale features of the solution are incorporated into these localized base functions which contain information about the scales that are smaller (as well as larger) than the local numerical scale defined by the base functions. A global formulation couples these base functions to provide an accurate approximation of the solution.

2.1 Governing Equations and the Multiscale Finite Element Method

The following second-order elliptic equation is considered.

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

where $a(x) = (a_{i,j}(x))$ is the conductivity tensor and is assumed to be symmetric and positive definite with upper and lower bounds. In practice, $a(x)$ maybe random or highly oscillatory; thus the solution of (2.1) displays a multiple scale structure.

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

$$a(u,v) = f(v) \quad \forall v \in H_0^1(\Omega), \tag{2.2}$$

where

$$a(u,v) = \int_\Omega a_{ij} \frac{\partial v}{\partial x_i} \frac{\partial u}{\partial x_j} dx, \quad f(v) = \int_\Omega f v dx$$

A finite element method is obtained by restricting the weak formulation (2.2) to a finite-dimensional subspace of $H_0^1(\Omega)$. For $0 < h \leq 1$, let $\mathbb{K}^h$ be a partition of Ω by a collection of K with diameter $\leq h$, which is defined by an axi-parallel rectangle mesh. In each element $K \in \mathbb{K}^h$, we define a set of nodal basis $\{\phi_K^i, i = 1, \cdots, d\}$ with d being the number of nodes of the element (for

2D domain, $d = 4$). The subscript K will be neglected when bases in one element is considered. In our multiscale method, ϕ^i satisfies

$$\bigtriangledown \cdot a(x) \bigtriangledown \phi^i = 0 \quad in\ K \in \mathbb{K}^h \tag{2.3}$$

Let $x_j \in \overline{K}(j = 1, \cdots, d)$ be the nodal points of K. As usual, we require $\phi^i(x_j) = \delta_{ij}$. One needs to specify the boundary condition of ϕ^i to make (2.3) a well-posed problem, which is discussed in next section.

The approximate solution of (2.1) in V^h, i.e., $u^h \in V^h$ is written as follows

$$a(u^h, v) = f(v) \quad \forall v \in V^h \tag{2.4}$$

2.2 The Boundary Condition of Base Functions

Boundary condition plays an important role in MsFEM since base functions satisfy the homogeneous equation (2.3). In fact, the boundary condition determines how well a local property of the property of the operator is sampled into the base functions.

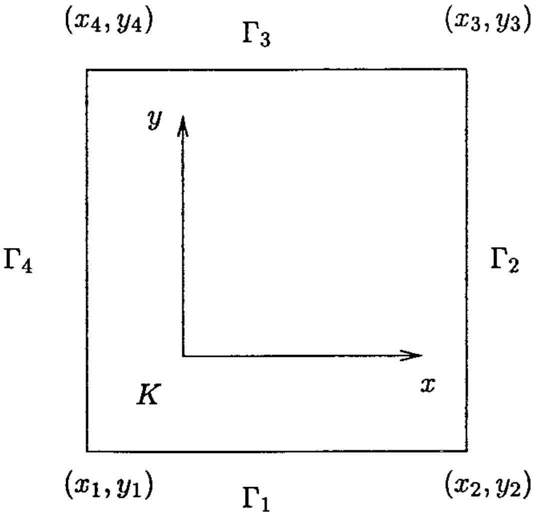


Figure 2.1

Denote $\mu^i = \phi^i|_{\partial K}$. An 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 (2.3) 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 of (2.3). When $a(x)$ is separable in space, i.e., $a(x) = a_1(x)a_2(y)$, 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. 2.1 [1]). Furthermore, it can be shown that this boundary condition is optimum for the space-separable problems.

To be more specific, consider an element $K \in \mathbb{K}^h$ with nodal points $x_i = (x_i, y_i)(i = 1, \cdots, d)$, which are labeled counterclockwise, starting from the lower left corner (Fig. 2.1)). On Γ_1 and Γ_3, we have $\mu^i = \mu^i(x)$ and

$$\frac{\partial}{\partial x} a_\mu(x) \frac{\partial \mu^i(x)}{\partial x} = 0, \tag{2.5}$$

where $a_\mu(x) = a_{11}|_{\Gamma_1}$ and $a_{11}|_{\Gamma_3}$, respectively. Similarly, on Γ_2 and Γ_4, we have $\mu^i = \mu^i(y)$ and

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

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

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

This can be seen as **general linear**. And if a_μ is a constant, then $\mu^1(x) = (x_2 - x)/(x_2 - x_1)$ is linear. In general, $\mu^i s$ 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. This is also verified in numerical experiments. Also, with this kind of boundary condition, one has

$$\sum_{i=1}^{d} \phi_K^i = 1 \quad \forall K \in \mathbb{K}^h \tag{2.8}$$

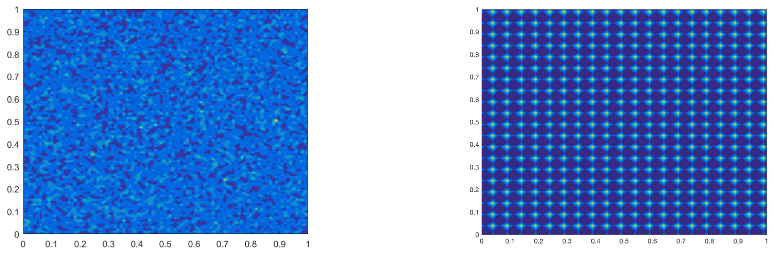
Thus, the constant functions belong to V^h.

3 Implementation and Numerical Experiments

In this section, we provide a detailed road map of implementing MsFEM. A numerical example is used to show that MsFEM could efficiently capture the large scale behavior of the solution without resolving all the small scale features. Compared with traditional finite element method, MsFEM could achieve the same level of accuracy with much less computational resource.

3.1 Problem Description

Many problem arising from engineering and scientific areas have multiple-scale solutions. Take elliptic problems in composite materials and porous media for instance, the media sometimes can be highly heterogeneous and $a(x)$ can be random (see Fig. 3.1a) or periodic (see Fig. 3.1b).



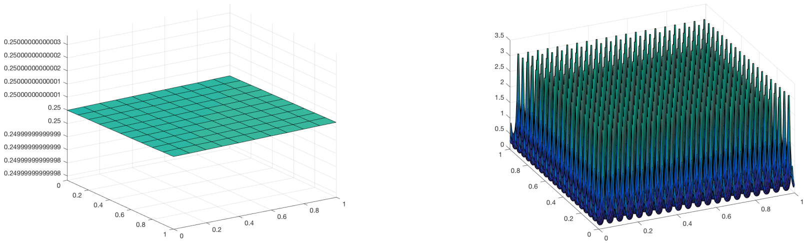
(a) Random $a(x)$

(b) Periodic $a(x)$

Figure 3.1: Contour plot of heterogeneous $a(x)$

In this project, we mainly focused on the periodic $a(x)$. The elliptic problem is defined on unit square domain ($(0,1) \times (0,1)$) with the following $a(x)$, $f(x)$ and zero boundary condition.

$$\begin{aligned} a(x) &= \frac{1}{(2 + Psin(2\pi x/\epsilon))(2 + Psin(2\pi y/\epsilon))} \\ f(x) &= -1, \quad u|_{\partial\Omega} = 0 \end{aligned} \tag{3.1}$$

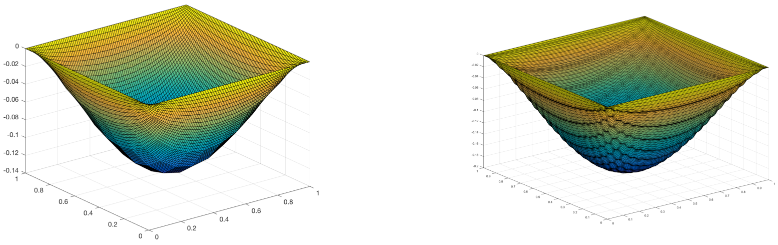


(a) 10 × 10 Sampling

(b) 200 × 200 Sampling

Figure 3.2: Different samplings of $a(x)$

Where $P = 1.5$. If the period ϵ is small (i.e., $\epsilon = 0.05$), the problem can be hard to solve with traditional FEM. Because in order to capture the oscillatory behavior of a, mesh used should be fine enough to resolve a, i.e., $h \ll \epsilon$. This introduces enormous degree of freedoms which makes the problem computationally prohibitive. We use an example to show why h has to resolve ϵ (See Fig. 3.2). Fig. 3.2a and Fig. 3.2b sample the same $a(x)$ ($\epsilon = 0.05$) with different sampling frequencies. On the left, the sampling frequency is half of $a(x)'s$ frequency ($h = 2\epsilon$). On the right, it is 10 times of the frequency of $a(x)$ ($h = 0.1\epsilon$). We can see the sample behavior is completely different with this extreme aliasing effect in the left figure, in which the solver is actually solving a problem with constant $a(x)$. So in order to obtain a reasonable solution, h must resolve $a(x)$.



(a) 10 × 10 Grid

(b) 200 × 200 Grid

Figure 3.3: Solving the problem with different meshing

We solved both discretization with traditional FEM using bilinear base functions and plotted the solution in Fig. 3.3. For $h = 0.1$, since it is essentially solving a Poisson equation with constant $a(x)$, we can not see any fluctuation in the solution. In comparison, using 200×200 gird ($h = 0.05$), the solution indeed reflects the fluctuation of $a(x)$. The problem with traditional FEM is the computational cost is high. In the above case, 200×200 grid can eat up 12 GB of memory. From this, we can see this kind of problem is hard for traditional FEM to solve.

3.2 Implementation

Since MsFEM given in Section 2 can be implemented within an existing finite element code, it is fairly straightforward to implement. We consider solving problem in a unit square domain. Let N be the number of elements in x and y directions. The coarse 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 $h = H/M$. Below, we present a simple pseudo-code that outlines the implementation of MsFEM.

Algorithm 3.1 Multiscale Finite Element Method

```
Mesh the domain
For each coarse element $K \in \mathbb{K}^h$ do
- For each node i
- Solve for $\phi_K^i$ satisfying $\bigtriangledown \cdot a(x) \bigtriangledown \phi_K^i = 0$ and boundary conditions (2.2)
- End for
Assemble stiffness matrix on the coarse mesh
Assemble the external force on the coarse mesh
Solve the coarse formulation
```

It should be noted that MsFEM is solved on the coarse mesh. Linear elements (with bilinear base functions) are used to solve the subcell problem for the base functions. If the coefficients a is differentiable and h resolves the smallest scale in a, then ϕ^i is computed with second order accuracy. The volume integrals

$$\int_K \bigtriangledown\phi^i \cdot a \cdot \bigtriangledown\phi^j dx \quad and \quad \int_K \phi^i f dx, \tag{3.2}$$

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. When $a(x)$ is separable in space, base functions can be computed analytically from tensor product of boundary conditions on adjacent edges. Also a map function that maps local node index to global node index is needed.

3.3 Base Function

In this section, we would like to show that how base functions used in MsFEM differ from that of traditional FEM. Traditional FEM uses bilinear base functions (Fig. 3.4) for 4 nodes rectangle elements. Bilinear base functions do not contain any information of media $a(x)$.

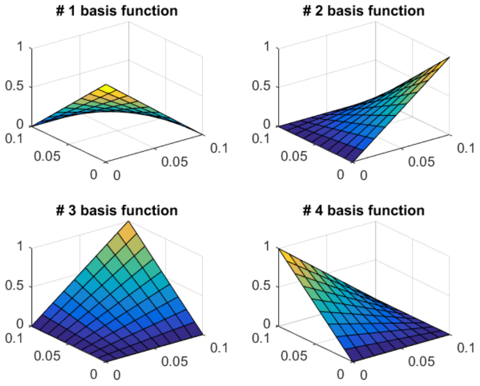


Figure 3.4: Bilinear base functions

In comparison, base functions (Fig. 3.5) used in MsFEM are different from bilinear base functions. Clearly they satisfy $\phi^i(x_j) = \delta_{ij}$ and continuity across element boundary. In addition, the small scale information is captured by the base functions constructed from solving the homogeneous equation (2.3). We also compare the boundary condition used in bilinear base functions and base functions of MsFEM (Fig. 3.6). It can be seen from Fig. 3.6 that unlike the linear boundary used in bilinear base functions, MsFEM's base functions have fluctuations.

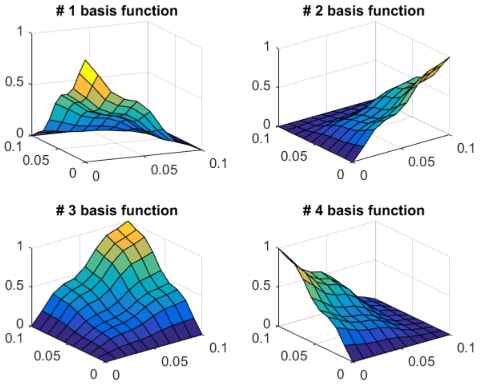


Figure 3.5: Base functions used in MsFEM ($\epsilon = 0.05$)

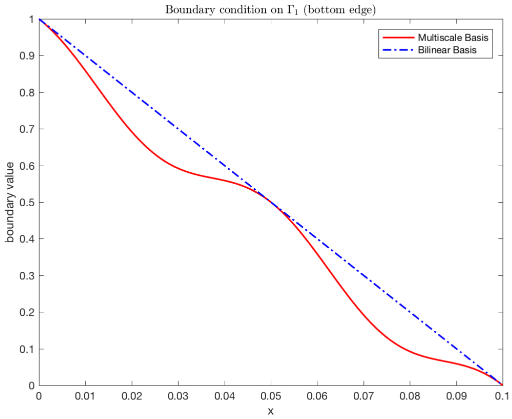


Figure 3.6: Boundary condition of base functions ($\epsilon = 0.05$)

Moreover we plot base functions calculated by a different $a(x)$ ($\epsilon = 0.01$). Compare Fig. 3.7 and Fig. 3.5, we can see that MsFEM's base functions vary according to $a(x)$.

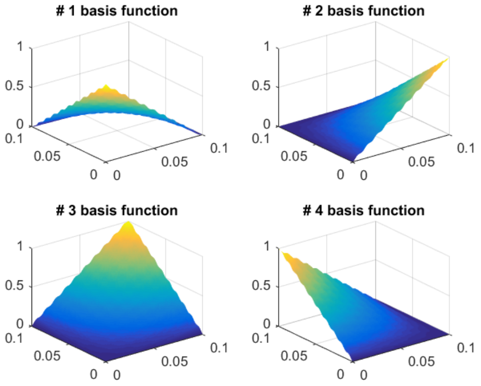


Figure 3.7: Base functions used in MsFEM ($\epsilon = 0.01$)

3.4 Numerical Results and Discussion

Apply both traditional FEM and MsFEM to problem (3.1) with $\epsilon = 0.05$, we can get the following results. For traditional FEM, in order to resolve $a(x)$, 200×200 grid is used. The solution is shown in Fig. 3.8. Since $h = 0.1\epsilon$, the obtained solution represents the oscillatory of the media.

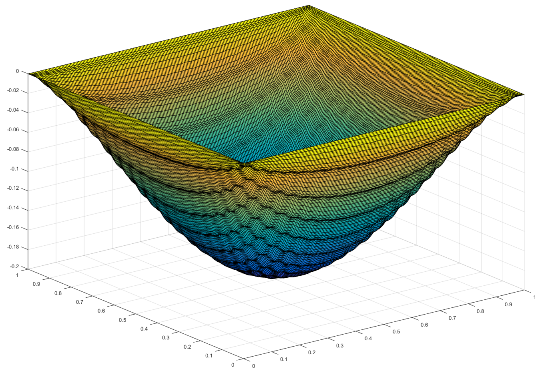


Figure 3.8: Traditional FEM solution (200 × 200 grid)

For MsFEM, we chose $H = 0.1$ and $h = 0.05$. The result is shown in Fig. 3.9. We can see the result obtained from MsFEM is nearly identical to that from traditional FEM with very fine grid. Our testing hardware is Intel 6700k CPU + 16 GB memory. Code is implemented using Matlab 2015. Solving the 200×200 traditional FEM took about 223.75 s with 11.92 GB memory footprint. However, solving the MsFEM is much more computationally efficient. For this problem ($a(x)$ is separable in space), we could use two methods to construct base functions. First one is using tensor product of μ^i along Γ_{i-1} and Γ_i. In this case it took about 6.93 s to solve the problem. The other method is using linear elements to solve the subcell problem for the base functions. In this case, it took about 27.01s. Both approaches used about the same amount of memory footprint (1.22 MB), which is much less than that used in traditional FEM.

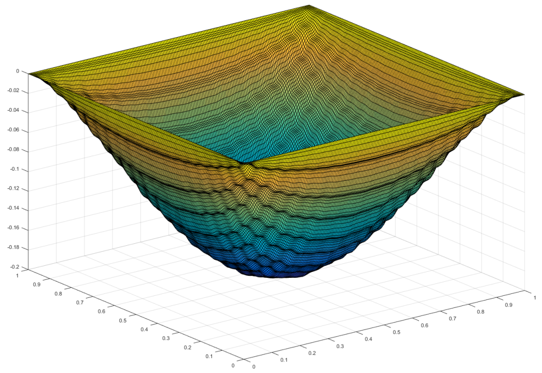


Figure 3.9: MsFEM solution ($H = 0.1$, $h = 0.05$)

Since $H = 2\epsilon$ and $a(x)$ is periodic, all elements share the same base function formulation. If $a(x)$ is not periodic or H is not integer times of ϵ, base functions should be calculated individually for each element. However the construction of the base functions is fully decoupled from element to element; thus MsFEM is perfectly parallel and is naturally adapted to massively parallel computers.

Another advantage of this 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 be the number of elements in each spatial direction, and let M be the number of subcell elements in each direction for solving the base functions. Then there are total $(MN)^n$ (n is the dimensions) elements at the fine grid level. For a traditional FEM, the computer memory needed for solving the problem on the fine grid is $O(M^nN^n)$. In contrast, MsFEM requires only $O(M^n + N^n)$ amount of memory.

4 Conclusion

In this report, we introduced the multiscale finite element method for elliptic problems with highly oscillatory media. We mainly focused on periodic media. Unlike traditional FEM, MsFEM captures the small scale information in base functions constructed from the leading order elliptic operator. The detailed implementation is described in Section 3.2 and 3.3. An elliptic problem with periodic media is used to show that MsFEM is consistent with the traditional finite element method in a well-resolved computation. We also demonstrated that MsFEM is much more computationally efficient and has much less memory footprint.

References

[1] Thomas Y Hou and Xiao-Hui Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics*, 134(1):169–189, 1997.

[2] JF McCarthy. Comparison of fast algorithms for estimating large-scale permeabilities of heterogeneous media. *Transport in Porous Media*, 19(2):123–137, 1995.

[3] Thomas Hou, Xiao-Hui Wu, and Zhiqiang Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Mathematics of Computation of the American Mathematical Society*, 68(227):913–943, 1999.

[4] Yalchin Efendiev and Thomas Y Hou. *Multiscale finite element methods: theory and applications*, volume 4. Springer Science & Business Media, 2009.