An iteratively adaptive multiscale finite element method for elliptic interface problems

Feng-Nan Hwang[a,*], Yi-Zhen Su[a], Chien-Chou Yao[a]

[a]*Department of Mathematics, National Central University, Jhongli District, Taoyuan City 32001, Taiwan*

Abstract

We develop and study a framework of multiscale finite element method (MsFEM) for solving the elliptic interface problems. Finding an appropriate boundary condition setting for local multiscale basis function problems is the current topic in the MsFEM research. In the proposed framework, which we call the iteratively adaptive MsFEM (i-ApMsFEM), the local-global information exchanges through iteratively updating the local boundary condition. Once the multiscale solution is recovered from the solution of global numerical formulation on coarse grids, which couples these multiscale basis functions, it provides feedback for updating the local boundary conditions on each coarse element. The key step of i-ApMsFEM is to perform a few smoothing iterations for the multiscale solution to eliminate the high-frequency error introduced by the inaccurate coarse solution before it is used for setting the boundary condition. As the method iterates, the quality of the MsFEM solution improves, since these adaptive basis functions are expected to capture the multiscale feature of the approximate solution more accurately. We demonstrate the advantage of the proposed method through some numerical examples for elliptic interface benchmark problems.

Keywords: Adaptive multiscale finite element basis, multiscale problem, elliptic interface problem, bubble functions.

1. Introduction

Multiscale finite element/finite volume methods [7, 8, 14, 15, 18, 19] have been demonstrated to be accurate and efficient numerical schemes for solving physical and engineering problems modeled by partial differential equations that exhibit multiscale phenomena. Contrary to classical Galerkin finite element methods using polynomial-type basis functions, typically lower order ones, such as linear or quadratic functions, the key ingredient of multiscale finite elements is the set of basis functions constructed locally by solving original partial differential equations restricted to the so-called *coarse element* with some proper boundary conditions. The multiscale feature of the solution is taken account through these basis functions in the global coarse formulation so that it provides a more accurate approximate solution when using much fewer grid points than the classical approaches.

It is known in the multiscale finite element literature that the choice of the boundary condition for the construction of multiscale basis functions plays an important role in the overall performance of the multiscale method. Generally speaking, an optimal boundary condition is often not known in *a priori* and is problem-dependent. Depending on the settings of the boundary conditions for the multiscale basis functions, the multiscale finite element methods can typically be classified into the following categories. The first one is to employ the local information purely. The simplest one is to use a linear boundary condition that connects two adjacent coarse nodal points, e.g., the residual-free bubble [9, 11, 12]. A more sophisticated

*Corresponding author. Tel: +886-3-422-7151 Ext. 65110; Fax +886-3-425-7379

Email addresses: hwangf@math.ncu.edu.tw (Feng-Nan Hwang), nick760110@gmail.com (Yi-Zhen Su), ccyao001@gmail.com (Chien-Chou Yao)

choice is to employ the solution of the original differential operator restricted to each edge of the coarse elements [13, 18]. However, if the selection of the boundary conditions is not able to reflect the nature of multiscale solution behavior, the error introduced by the boundary condition will cause a large error in the global coarse solution. In [18, 19] the two-scale analysis based on the homogenization theory revealed that the convergence rate of the multiscale method degrades mainly due to boundary resonance error between the coarse element size and the characteristic length. The problem becomes severe for the case that these two quantities are close to each other. One remedy solution is to use the oversampling technique [18, 19], where the multiscale basis functions are constructed by using the information on the overlapped subdomains in order to reduce the boundary effects. The second class is to employ the global solution information. One can solve a simplified model problem and use its solution on whole domain for setting the boundary condition for multiscale basis problem. For example, in the two-phase flow simulation, Efendiev *et al.* [7] demonstrate the advantage of using the projection of the global fine-scale solution of the single-phase equation at the initial time for setting the boundary condition numerically.

Interface problems arise from a variety of applications in computational science and engineering [3, 5, 20]. For the elliptic interface problems, the convergence rate of the standard Galerkin finite element method is shown, either numerically or theoretically, to be suboptimal [23] if the finite element nodal points cannot resolve the interface curve. This is often the case in many practical applications, and a body-fitting mesh is usually expensive to construct. In this paper, we study a new framework of MsFEM named "iteratively adaptive multiscale finite element methods" (or i-ApMsFEM) for solving elliptic interface problems [4, 5, 21, 23]. In the proposed framework, the local-global information iteratively exchanges through the updated local boundary condition for these multiscale basis problems. Once the multiscale solution is recovered from the solution of the global numerical formulation on coarse grids, which couples these multiscale basis functions, it provides feedback for updating the local boundary conditions on each coarse element. As we increase the number of iterations, the quality of MsFEM solutions improves, since these adaptive basis functions are expected to be able to capture the multiscale feature of the approximate solution more accurately. The key step of i-ApMsFEM is to perform a few steps of smoothing iterations for the multiscale solution before it is used for setting the boundary condition for the multiscale basis functions in order to eliminate the high-frequency error introduced by the inaccurate coarse solution. Alternatively, the other class of iterative multiscale finite volume methods was recently developed by Jenny and his coworkers [14, 15]. It can be viewed as the classical multiscale finite element space enriched by a space spanned by adaptive bubble-shaped basis functions, in which both of multiscale basis functions and bubble-shaped basis functions are defined on the dual coarse cells. Then the multiscale solution is recovered on the primary coarse grid. Unlike residual-free bubble functions requiring the zero values on all boundaries, the bubble type basis functions in this method are iteratively updated. The bubble functions along the boundary are allowed to take nonzero values depending on the global solution. Only the values of the bubble functions at the nodal points are kept to be zero. Other similar classes of iterative MsFEMs were proposed by Millward [23]. It can be viewed as a generalization of oversampled MsFEM, in which the boundary values for the local multiscale basis functions are adaptively updated via the data exchange on the overlapping regions along the boundary curve.

The rest of the paper is organized as follows. In Section 2, we describe a general framework of the proposed iteratively adaptive finite element methods for solving the second order PDEs. In Section 3, we present the numerical results for the high-contrast interface problem, and we give some concluding remarks in Section 4.

2. A description of i-ApMsFEM method for a PDE model problem

2.1. Model problem

Let Ω be an open and bounded domain with $\partial\Omega$ in $\mathbb{R}^2$. For simplicity, we assume that $\partial\Omega$ is a polygonal curve, in which case we say that Ω is a polygonal domain. (If $\partial\Omega$ is a curve, we can approximate it with a polygon). Consider the following model problem,

$$\begin{cases} \mathcal{L}u = f \\ u = g, \end{cases} \tag{1}$$

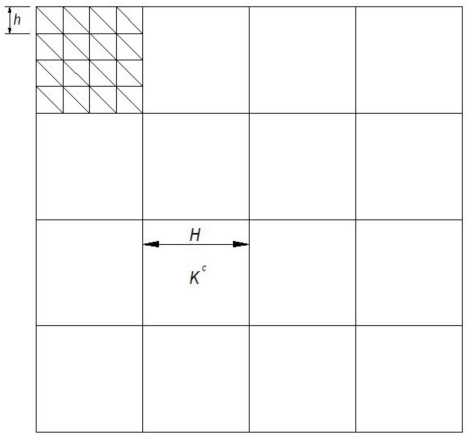

Figure 1: A sample of coarse element and fine element

where $\mathcal{L}$ is a second-order linear elliptic PDE operator with a discontinuous coefficient, u is an unknown scalar function, g is a prescribed Dirichlet-type datum, and f is a given source function. We also assume that this problem is well posed.

2.2. Adaptive multiscale finite element formulation

Next, we describe i-ApMsFEM in detail as follows. As shown in Figure 1, consider the quadrilateral coarse mesh $\mathcal{T}^H = \{K^c\}$ with an element size H. For each K^c, we further partition it into fine meshes $\mathcal{T}^h = \{K\}$ with an element size h for the purpose of the numerical construction of multiscale finite element basis and bubble functions on the coarse elements. For simplicity, we assume the interfacial nodal points between neighboring coarse elements match each other. Let s be the current iteration number and $s_{\max}$ be the maximum number of iterations allowed. Define an adaptive multiscale finite element space $M_H^{(s)}$ as

$$M_H^{(s)} = span\{\varphi_i^{(s)}, i = 1, ..., N_{np}^c\}.$$

Here, $\varphi_i^{(s)}$ is denoted by the global multiscale finite element basis, which is updated as the method iterates and N_{np}^c is the total number of coarse nodal points. The computation of $\{\varphi_i\}$ will be discussed explicitly later in next subsection. The i-ApMsFEM formulation reads as: For $s = 1, ..., s_{\max}$, find $u_H^{(s)} \in M_H^{(s)}$ such that

$$B(u_H^{(s)}, v_H) = L(v_H) \text{ for } \forall v_H \in M_H^{(s)}, \tag{2}$$

where $B(\cdot, \cdot)$ is a bilinear form and $L(\cdot)$ is a linear form. For $u_H^{(s)} \in M_H^{(s)}$, it can be written as

$$u_H^{(s)} = \sum_{j=1}^{N_{np}^c} u_j^{(s)} \varphi_j^{(s)}, \tag{3}$$

where $u_j^{(s)}$ is the *the nodal value of the muitiscale solution on the coarse grid* at the s-th iteration. Then, each multiscale basis function $\varphi_j^{(s)}$ can be further expressed as a linear combination of the finite element basis functions $\{\psi_k\}$ defined on the fine grid, i.e.,

$$\varphi_j^{(s)} = \sum_{k=1}^{N_{np}^f} d_{jk}^{(s)} \psi_k,$$

where N_{np}^f is the total number of fine nodal points. Note that $\{\psi_k\}$ can be chosen differently depending on application such as standard linear finite element basis, or immersed finite element basis [20]. By choosing $v_H = \varphi_i^{(s)} = \sum_{l=1}^{N_{np}^f} d_{il}^{(s)} \psi_l$ and substituting Eq. (3) into the weak formulation (2), we obtain the coarse grid problem,

$$\sum_{j=1}^{N_{np}^c} \left[\sum_{k=1}^{N_{np}^f} \sum_{l=1}^{N_{np}^f} d_{jk}^{(s)} B(\psi_k, \psi_l) d_{il}^{(s)} \right] u_j^{(s)} = \sum_{l=1}^{N_{np}^f} d_{il}^{(s)} L(\psi_l), \tag{4}$$

for $i = 1, ..., N_{np}^c$ or more compactly in the matrix form as

$$D^{(s)} A^f \left(D^{(s)}\right)^T \mathbf{x}_c^{(s)} = D^{(s)} b_*^f,$$

where $A^f = (A_{kl}^f)$ and $b^f = (b_i^f)$ with $A_{kl}^f = B(\psi_k, \psi_l)$ and $b_l^f = L(\psi_l)$, respectively. Besides, the matrix $D = (d_{kl})$ is called the multiscale basis function matrix, that transfers data from the fine grid of the dimension N_{np}^f to the coarse grid of the dimension N_{np}^c, is updated during the iteration and $\mathbf{x}_c = (u_1, u_2, ..., u_{N_{np}^c})^T$ is the multiscale solution vector on the coarse grid.

2.3. Construction of multiscale basis functions

By introducing the local multiscale finite element basis $\{\phi_m\}$ corresponding to the global ones, restricted to the coarse element K^c, each ϕ_m is defined to satisfy the following homogeneous boundary value problem,

$$\begin{cases} \mathcal{L}\phi_m = 0 \text{ in } K^c, \\ \phi_m = \mu^{(s)} \text{ on } \partial K^c \\ \phi_m(\mathbf{p}_n) = \delta_{mn} \text{ for } n = 1, ..., N_{en}^c \text{ and } \mathbf{p}_n \text{ are the local coarse nodal points.} \end{cases} \tag{5}$$

Here, N_{en}^c is the number of local coarse element nodes.

To get the iterative process started, the linear interpolation connecting two local coarse points to satisfy the last condition in (5) is used as an *initial* setting of the boundary condition, $\mu^{(s=0)}$. As indicated from the numerical results in Section 3, the error introduced by the setting of the boundary condition is soon to be corrected within the first few iterations. For the case of $u_H^{(s-1)}(\mathbf{p}_m) = u_H^{(s-1)}(\mathbf{p}_n) = 0$ or $(\neq 0)$, we simply take μ as a linear function along ∂K^c. On the other hand, for the case of $u_H^{(s-1)}(\mathbf{p}_m) \neq u_H^{(s-1)}(\mathbf{p}_n)$, we set

$$\mu^{(s)}(\mathbf{p})|_{\partial K^c} = \frac{u_H^{(s-1)}(\mathbf{p}) - u_H^{(s-1)}(\mathbf{p}_m)}{u_H^{(s-1)}(\mathbf{p}_n) - u_H^{(s-1)}(\mathbf{p}_m)} \text{ for } \mathbf{p}_m \text{ and } \mathbf{p}_n \text{ sharing the common edge,}$$

where $u_H^{(s-1)}$ is the multiscale solution at the $(s-1)$th iteration, whose nodal values on a fine grid can be recovered as

$$\mathbf{x}_f^{(s-1)} = \left(D^{(s-1)}\right)^T \mathbf{x}_c^{(s-1)}.$$

The following observation motivates such boundary condition settings: if u_H is replaced by the exact solution u on the coarse nodal points, the function $\mu|_{\partial K}$ will recover the exact solution u along the edge. In practice, the exact solution is not available. However, we will see later, in the numerical result section that the proposed i-ApMsFEM algorithm itself is an efficient way of finding the approximate fine-grid solution $\mathbf{x}_f$. Before $\mathbf{x}_f^{(s-1)}$ can be used for the computation of the boundary condition for the local multiscale basis function, as suggested by [14, 15], we perform a few smoothing operations by using some iterative methods with the smoothing property [24]. Some possible options include a weighted Jacobi iteration, incomplete LU decomposition with no fill-ins (ILU(0)), and a preconditioned Krylov subspace method. The roles of this smoothing step are to eliminate the oscillatory boundary error efficiently due to an inaccurate approximate fine-scale solution and to enhance the global communications between the coarse elements. Note that, in our construction, if there were no smoothing iteration, our iteration would remain the same after the first step. Using the ILU(0) decomposition method as a concrete example, the ν-ILU(0) smoothing steps can be

described as follows: Let $A^f = \hat{\mathcal{L}}\hat{\mathcal{U}} - \mathcal{R}$, where the matrix A^f represents the global matrix obtained by some proper numerical schemes, e.g., immersed, stabilized, or Galerkin finite element methods on the fine grid, $\hat{\mathcal{L}}$ and $\hat{\mathcal{U}}$ are an incomplete LU decomposition of A^f. Given $\mathbf{y}^{(0)} = \mathbf{x}_f^{(s-1)}$, for $k = 1, ..., \nu$

$$\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)} + \mathbf{z}^{(k-1)},$$

where $\mathbf{z}^{(k-1)}$ is obtained by solving

$$\hat{\mathcal{L}}\hat{\mathcal{U}}\mathbf{z}^{(k-1)} = \mathcal{R}\mathbf{y}^{(k-1)} + b_f^*$$

by using forward/backward substitutions.

In general, the analytical solutions to the local problems (5) are usually not available; however they can be approximated by some proper numerical scheme at the fine-scale level, for example, the immersed finite elements for the interface problems [16, 21], the stabilized finite elements for the convective-diffusive problems [6, 10, 25] and Helmholtz equation [1], or simply Galerkin finite elements for the elliptic problems with the multiscale diffusion coefficient [18, 19]. These numerical schemes for the local problem (5) are written in the matrix formulation, for $i = 1, ..., N_{np}^c$,

$$R_i A^f R_i^T \Phi_m^i = b_\mu^{(s)},$$

where R_i and R_i^T are the restriction and interpolation matrices mapping data from the global space in to the local space, and from the local space to the global space, respectively. $b_\mu^{(s)}$ contains all updated information due to the changes of boundary condition for the local multiscale basis functions. Here, the vector Φ_m^i contains the nodal values of the local multiscale basis ϕ_m^i.

2.4. The case when the external forcing term presents

For the inhomogeneous case, i.e. $f \neq 0$, the bubble function space, B_H needs to be introduced, and the trial function spaces for this case are the multiscale function space at each iteration s enriched by the bubble function spaces, i.e.,

$$M_H^{(s)} \oplus B_H.$$

The bubble function restricted to each coarse element φ_i^B satisfies the following inhomogeneous boundary value problem, for $i = 1, ..., N_{el}^c$,

$$\begin{cases} \mathcal{L}\varphi_i^B = f \text{ in } K^c, \\ \varphi_i^B = 0 \text{ on } \partial K^c. \end{cases}$$

and the corresponding discrete problem with some proper discretization is given

$$R_i A^f R_i^T \mathbf{\Phi}_i^B = R_i b_f.$$

As a result, the fine grid solution then is obtained via

$$\mathbf{x}_f = D^T \mathbf{x}_c + \sum_{i=1}^{N_{el}^c} R_i^T \mathbf{\Phi}_i^B,$$

where $D^T \mathbf{x}_c \in M_H^{(s)}$ and $\sum_{i=1}^{N_{el}^c} R_i^T \mathbf{\Phi}_i^B \in B_H$. Note that the bubble function is independent of the i-ApMsFEM solution, hence it can be precomputed before entering the iterative process.

2.5. A completed description of the i-ApMsFEM algorithm

The complete algorithm for iterative multiscale finite elements is summarized in Algorithm 1.

Algorithm 1 Iteratively adaptive multiscale finite element algorithm

for $s = 1$ to $s_{\max}$ **do**
  for $i = 1$ to N_{el}^c **do**
    for $m = 1$ to N_{en}^c **do**
      Compute the boundary condition for multiscale basis function problems.

For $\mathbf{p}_m$ and $\mathbf{p}_n$ sharing the common edge, we consider

Case I: If $u_H^{(s-1)}(\mathbf{p}_m) = u_H^{(s-1)}(\mathbf{p}_n) = 0$ or $(\neq 0)$ then
μ as a linear function along ∂K^c.

Case II: If $u_H^{(s-1)}(\mathbf{p}_m) \neq u_H^{(s-1)}(\mathbf{p}_n)$ then

$$\mu^{(s)}(\mathbf{p})|_{\partial K^c} = \frac{u_H^{(s-1)}(\mathbf{p}) - u_H^{(s-1)}(\mathbf{p}_m)}{u_H^{(s-1)}(\mathbf{p}_n) - u_H^{(s-1)}(\mathbf{p}_m)},$$

where $n = 1, ..., N_{en}^c$.

Otherwise, set $\mu(\mathbf{p})|_{\partial K^c} = 0$

      Solve the multiscale basis function problems,

$$(R_i A^f R_i^T)\mathbf{\Phi}_m^i = \tilde{R}_i \hat{\mathbf{u}}_f,$$

      Update the multiscale function matrix, $D^{(s)}$
    end for
  end for
  Solve the coarse problem, $A^c \mathbf{x}_c^{(s)} = D^{(s)} \mathbf{b}_f$, where $A^c = D^{(s)} A^f (D^{(s)})^T$.
  Check if the user-defined stopping condition is satisfied, stop.
  Recover $\mathbf{x}_f^{(s)} = (D^{(s)})^T \mathbf{x}_c^{(s)}$
  Relax $A^f \hat{\mathbf{x}}_f = \mathbf{b}_f$, ν times with the initial guess, $\mathbf{x}_f^{(s)}$
end for

Remark 1. The framework of the i-ApMsFEM algorithm can be viewed as two-grid cycles scheme in the terminology of the multigrid [2, 26], and it can be used to serve as an iterative solver for large sparse linear discrete PDE problem on the fine grid. If we reformulate the coarse grid correction (Step 9 in Algorithm 1) as two steps:

$$\begin{aligned} A^c \mathbf{e}_c^{old} &= Dr_f \equiv D(\mathbf{b}_f - A^f \mathbf{x}_f^{old}) \\ \mathbf{x}_c^{new} &= \mathbf{x}_c^{old} + \mathbf{e}_c^{old}, \end{aligned}$$

the one-step i-ApMsFEM algorithm can be understood as

$$\mathbf{x}_f^{new} = S^\nu[(I - D^T(A^c)^{-1}DA^f)S^\nu \mathbf{x}_f^{old} + (D^T(A^c)^{-1}D)\mathbf{b}_f,$$

where S^ν is a ν-step smoothing operator. Hou et al. [17] carried out the convergence analysis of an algorithm equivalent to Algorithm 1 under the framework of the two-grid cycle method, where the approximation property of the adaptively constructed multi-scale basis functions to the solution space of equation is established by using the harmonic coordinates .

Remark 2. The major components of the i-ApMsFEM algorithm consist of (1) the construction of the multiscale basis matrix D (a set of local solves needed), (2) the formation and the solution of the coarse grid problem, and (3) the global smoother. Assume that both of the local problems and the coarse grid problem are solved by the dense LU decomposition and the forward/backward substitutions, which require $O(n^3 + n^2)$ operations and ILU(0) as a global smoothing operation, which in turn requires $O(2N_Z)$. Here, n is the size of the linear problem and N_Z is the total number of the nonzero elements of the problem. For our application, it is roughly $N_Z = 5n$. Consider a 2D problem defined on a unit squared domain. Assume that N is the number of coarse elements in each dimension. For each coarse element, it is further partitioned into $M \times M$ fine elements. Hence, $L = M \times N$ is the total number of elements in both directions. Let I be the number of the i-ApMsFEM iterations, which is relatively small compared to the problem size, and d be the number of the multiscale basis functions per coarse element. Therefore, the total cost of constructing the multiscale basis matrix D is as follows:

$$O((d-1)(M^6 + IM^4)N^2).$$

Also, the total costs of formulating and solving the coarse grid problems are

$$O(16IM^4N^2)$$

and

$$O(N^6 + IN^4)),$$

respectively, and the total cost of performing the ν smoothing step is

$$O(10\nu I(MN)^2).$$

For simplicity, we assume that the variables M and N are in the same order. Then the ratio of the extra cost of i-ApMsFEM compared to the classical MsFEM and the dominated cost for both MsFEMs is about $1/N^2$. In addition, the ratio of the computational complexity for solving i-ApMsFEM compared to the standard FEM on the fine grid is roughly about $1/N^4$, where the dense LU decomposition method is used for solving the resulting linear system arising from the standard FEM.

Remark 3. The main difference between our i-ApMsFEM algorithm and the one proposed by Millward [23] is the way to update the boundary condition for the local multiscale basis functions. Instead of using a global smoothing operator, Millward introduces two auxiliary sets of local mutltiscale basis functions. The first set of local multiscale basis functions is defined on a quadrilateral overlapping coarse element, i.e.,

$$\begin{cases} \mathcal{L}\tilde{\phi}_m^M = 0 \text{ in } \tilde{K}^c, \\ \tilde{\phi}_m^M = \mu^{(s)} \text{ on } \partial\tilde{K}^c \\ \tilde{\phi}_m^M(\mathbf{p}_n) = \delta_{mn} \text{ for } n = 1, ..., N_{en}^c \text{ and } \mathbf{p}_n \text{ are the coarse nodal points.} \end{cases} \tag{6}$$

Here, $\tilde{K}^c$ is an overlapping coarse element obtained from the extension of the non-overlapped one, i.e., $K^c \subset \tilde{K}^c$. Global information is used to set the boundary conditions, μ, for multiscale basis functions as we do along the boundary segments of $\tilde{K}^c$ (Step 4 in Algorithm 1). Once these basis functions are constructed, then the basis functions on the non-overlapping coarse element K^c are computed using a linear combination of the previous set of basis functions,

$$\phi_m^M(\mathbf{p}) = \sum_{i=1}^{N_{en}^c} c_{m,i}\tilde{\phi}_i^M(\mathbf{p}) \text{ for } m = 1, ..., N_{en}^c,$$

where the coefficient $c_{m,i}$ is determined by using the condition, $\phi_m^M(\mathbf{p}_n) = \delta_{m,n}$ for $m, n = 1, ..., N_{en}^c$ and $\mathbf{p}_n$ is the vertices of K^c. Note that the second set of the basis functions is nonconforming. To obtain a conforming one, one constructs another set of multiscale basis by solving the local problem (6) again but now on the non-overlapping coarse element K^c with the average value of two neighboring basis functions along an edge as the boundary condition, i.e.,

$$\mu \equiv \psi_i^M|_e = \frac{\phi_s^M|_e + \phi_t^M|_e}{2},$$

where $\phi_s^M|_e$ and $\phi_t^M|_e$ are any two basis functions sharing edge e between two adjacent coarse elements. Hence, the overhead of the Millward's approach for the construction of the adaptive multiscale basis is about twice as the one of the i-ApMsFEM algorithm.

3. Numerical results

We consider a series of benchmark problems to evaluate the performance of our proposed approach for solving the elliptic interface problem.

3.1. Test cases for elliptic interface problems

We first consider a problem with an analytical solution, referred to as the single inclusion problem [5, 23]. Let $\Omega = [-1, 1] \times [-1, 1]$. The elliptic interface problem is given by

$$\begin{cases} \mathcal{L}u \equiv -\nabla \cdot (\beta \nabla u) = f(x, y) \ (x, y) \in \Omega, \\ u|_{\partial\Omega} = g(x, y) \end{cases} \tag{7}$$

where the jump coefficient β is defined as

$$\beta = \begin{cases} \beta^-, & (x, y) \in \Omega^- \\ \beta^+, & (x, y) \in \Omega^+ \end{cases}$$

and two additional jump conditions along the interface Γ are imposed:

$$\begin{aligned} [u]_\Gamma &= 0, \\ [\beta \frac{\partial u}{\partial n}]_\Gamma &= 0. \end{aligned}$$

Here, the computational domain Ω is divided into two disjoint sub-regions Ω^+ and Ω^- such that $\Omega = \Omega^+ \cup \Omega^- \cup \Gamma$. The force term is given as $f = -9(x^2 + y^2)^{1/2}$ and the inhomogeneous Dirichlet-type boundary condition is set to be $g = \frac{(x^2+y^2)^{3/2}}{\beta^-} + (\frac{1}{\beta^-} - \frac{1}{\beta^+})r_0^3$ with $r_0 = \pi/6.28$. The exact solution is

$$u(x, y) = \begin{cases} \frac{(x^2+y^2)^{3/2}}{\beta^-} & \text{if} \quad \sqrt{x^2 + y^2} < r_0 \\ \frac{(x^2+y^2)^{3/2}}{\beta^+} + (\frac{1}{\beta^-} - \frac{1}{\beta^+})r_0^3 & \text{if} \quad \sqrt{x^2 + y^2} > r_0. \end{cases}$$

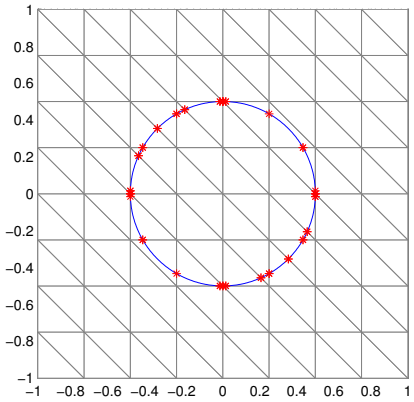

Figure 2: A sample uniform triangular Cartesian finite element mesh, where the interface curve is in blue and all interface points are labeled the red dots.

Figure 2 shows a sample uniform triangular Cartesian finite element mesh. Here, the red dots are the intersection points (or called the interface points) between the interface curve and the edge of finite element mesh. Two additional test examples, namely the double inclusions problem and the lens problem [23], are considered. For both cases, the computational domain Ω, external force term, f, and boundary condition, g are are chosen in the same way as in the previous single inclusion problem. As shown in Figure 3 for the double inclusions problem, the two sub-regions are defined as

$$\Omega^+ = \{(x,y) \in \Omega \mid \|(x,y) - c_1\| < r_1 \ \cup \ \|(x,y) - c_2\| < r_2 \ \} \text{ and } \Omega^- = \Omega \backslash \Omega^+,$$

where $c_1 = (0, -0.5),\ c_2 = (0, 0.5)$ and $r_1 = r_2 = \frac{\pi}{12.56}$, for the lens problem, we set

$$\Omega^+ = \{(x,y) \in \Omega \mid \|(x,y) - c_1\| < r \ \cap \ \|(x,y) - c_2\| < r \ \} \text{ and } \Omega^- = \Omega \backslash \Omega^+,$$

where $c_1 = (0, -c_y),\ c_2 = (0, c_y)$ with $c_y = \frac{\pi}{6.28}\sqrt{(2+\sqrt{2})/(2-\sqrt{2})}$, and $r = \frac{\pi}{6.28}\sqrt{4/(2-\sqrt{2})}$.

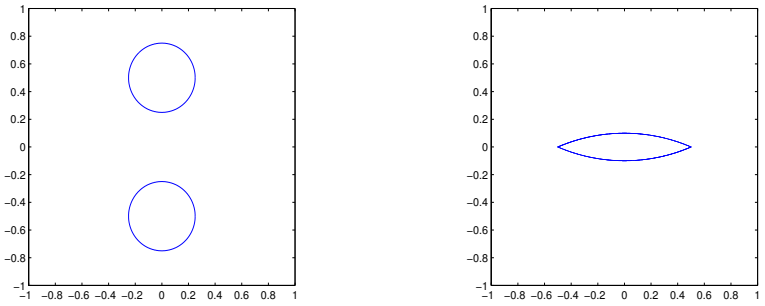

Figure 3: Double inclusions problem (left) and Lens problem (right)

In the following numerical experiments, a set of uniform rectangular coarse elements is used with the element size $H = 1/N$, where N is the number of coarse elements in the x- or y-direction. For the local problems in which we compute the multiscale finite element basis, the immersed finite element method [22] is used with a uniform *fine* triangular element with the element size $h = (1/M)H$. The same method is also used for constructing the global matrix defined on the fine grid, A^f as shown in Step 9 of Algorithm 1.

3.2. A comparison of initial and final MsFEM solutions

Figures 4 and 5 show a comparison of the initial and final multiscale finite element solutions for the single inclusion problem and the corresponding pointwise errors. In addition, Figures 6 and 7 display a comparison of initial and final MsFEM pointwise errors for the double inclusions problem and the lens problem, respectively. Two different values of the ratio $\hat{\alpha} = 10^{-3}$ and $\hat{\alpha} = 10^{3}$ are considered. Since there is no analytical solution available for these two test problems, the interface finite element solution with $h = 1/512$ is used as the reference solution to compute the pointwise errors. From this series of figures, some observations are made as follows.

1. For both cases, the *initial* MsFEM pointwise errors with large magnitude appear near or on the interface curve. The wave-like error is near the physical boundary for the case of $\hat{\alpha} = 10^{-3}$ due to the incorrect boundary condition for the local problem, but it is soon eliminated as the method iterates.
2. Generally speaking, for the single inclusion problem, the *final* MsFEM pointwise error near the interface region is reduced by about three orders of magnitude compared to the initial one. At least one order of magnitude reductions is also observed for the other two problems.

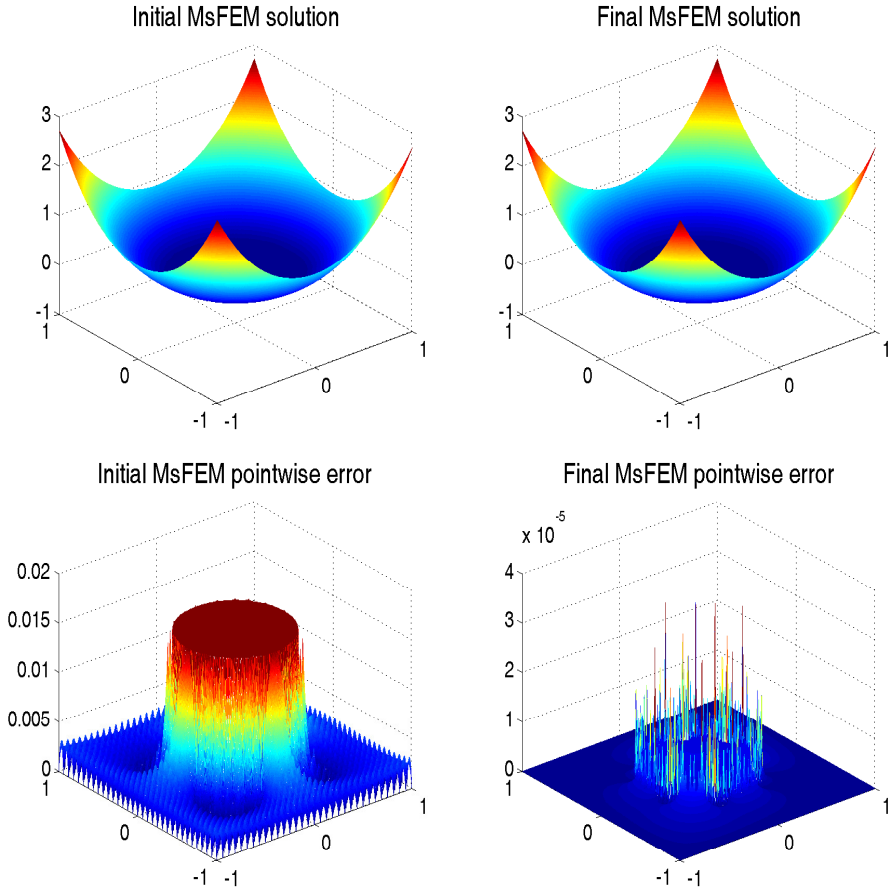

Figure 4: Single inclusion problem with $\hat{\alpha} = 10^{-3}$. Comparison of initial and final MsFEM solutions and pointwise errors. $N = 32$ and $M = 16$ are used.

3.3. Efficiency studies of i-ApMsFEM algorithm

Figure 8 shows a typical convergence history of the i-ApMsFEM algorithm. The L_2-error norm (red curves) is reduced significantly at the beginning as the number of iterations increases and then the error

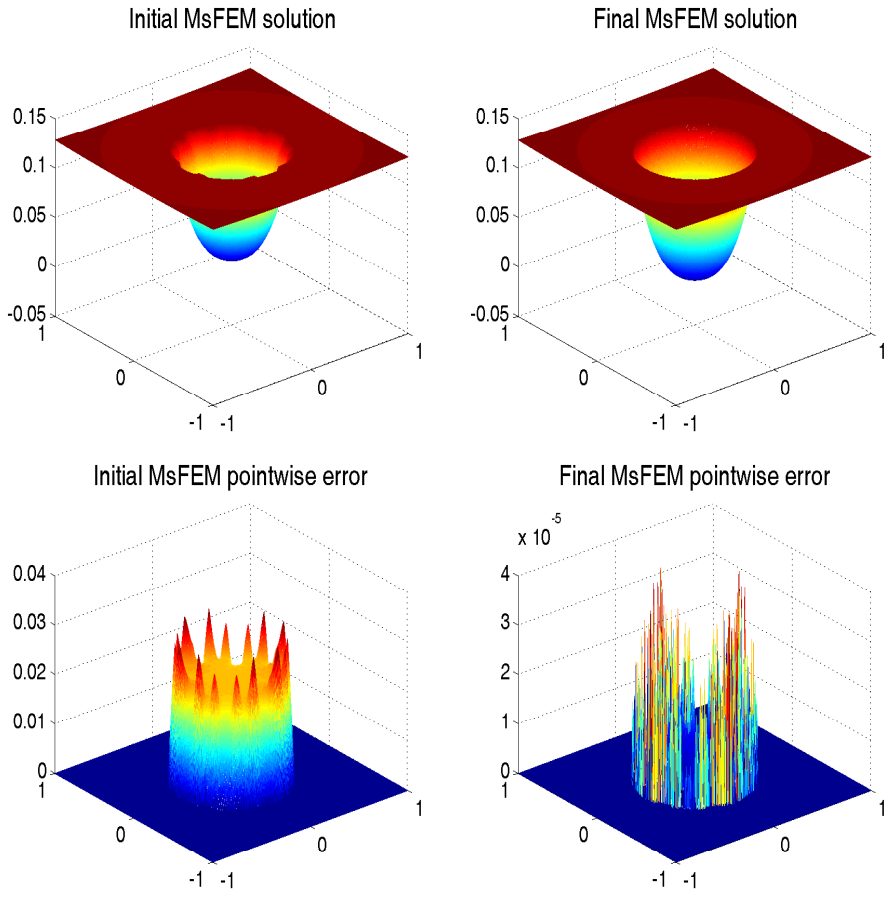

Figure 5: Single inclusion problem with $\hat{\alpha} = 10^3$. Comparison of initial and final MsFEM solutions and pointwise errors. $N = 32$ and $M = 16$ are used.

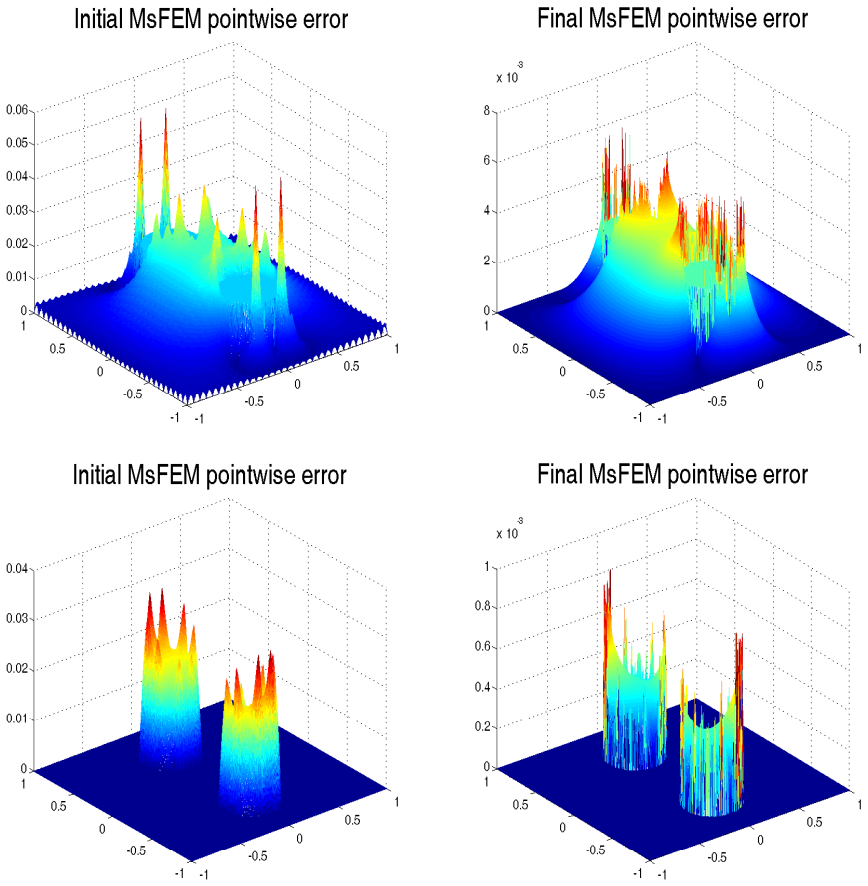

Figure 6: Double inclusions problem with $\hat{\alpha} = 10^{-3}$ (top row) and $\hat{\alpha} = 10^{3}$ (bottom row) . Comparison of initial and final MsFEM pointwise errors. $N = 32$ and $M = 16$ are used.

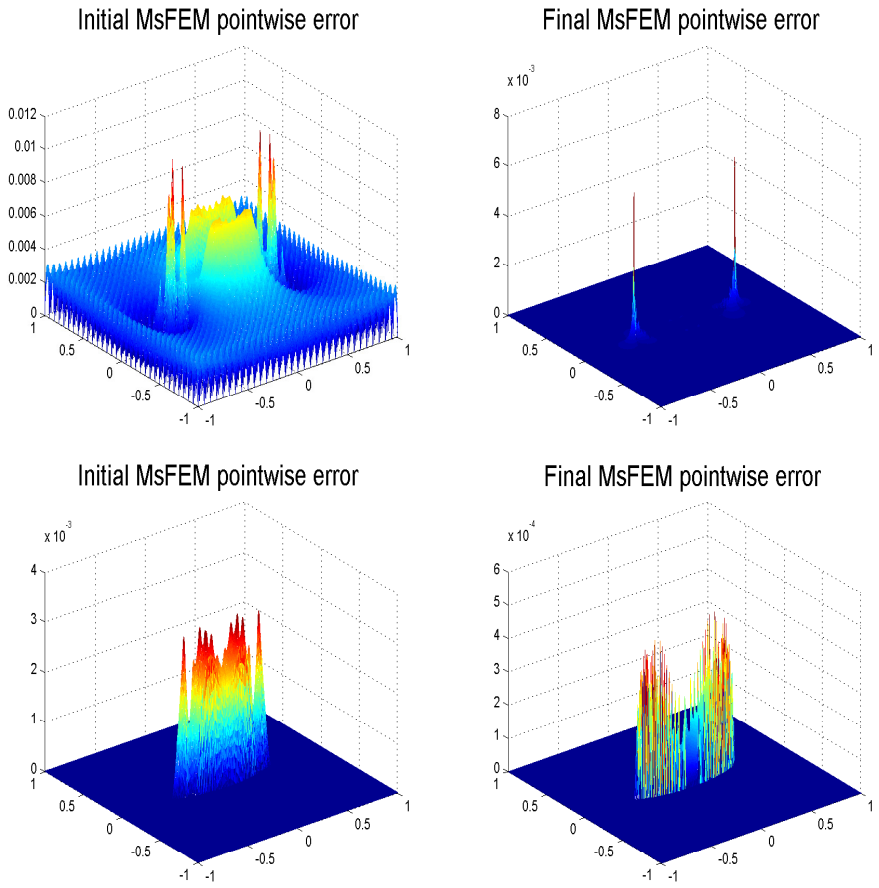

Figure 7: Lens problem with $\hat{\alpha} = 10^{-3}$ (top row) and $\hat{\alpha} = 10^{3}$ (bottom row) . Comparison of initial and final MsFEM pointwise errors. $N = 32$ and $M = 16$ are used.

stalls when it reaches the level of the truncation error. Here, one-step ILU(0) type smoother is employed. This figure suggests that the iterative process should be stopped earlier; at most ten iterations for this particular case are sufficient. Hence, a proper stopping condition and a mechanism to monitor the progress of the numerical solution toward to the desired solution are needed. Here we consider two possibilities: One is the difference between two consecutive MsFEM solutions on a coarse grid, denoted by $\|x_c - \text{pre } x_c\|_{L_2(\Omega)}$. The other one is based on the residual norm of the MsFEM solution on the fine grid, i.e., $\|b^f - A^f x_f\|_2$, where the coefficient matrix A_f and the right-hand side vector b_f come from the discretization of the immersed finite method on the fine grid. The trend of the curves (blue curves) in Figure 8 corresponds to the first difference norm monitor and follows the error reduction more closely than the second one does. Hence, $\|x_c - \text{pre } x_c\|_{L_2(\Omega)} < 10^{-6}$ will be used as the stopping condition. When this stopping condition is satisfied, i-ApMsFEM is considered to have converged in the following experiments.

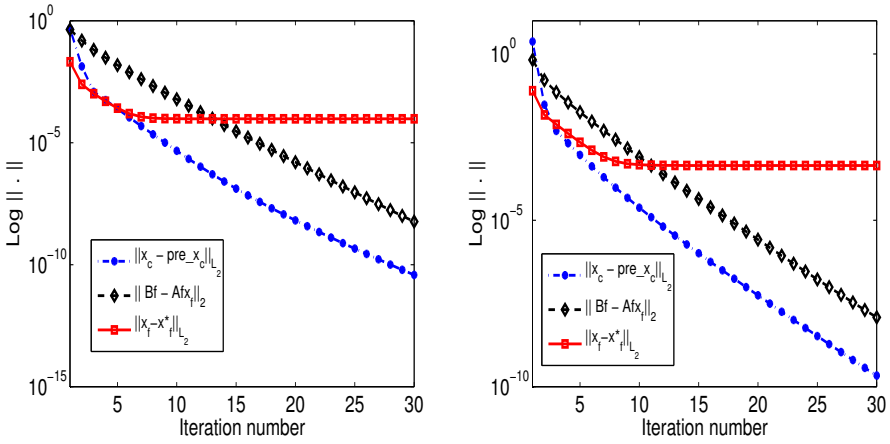

Figure 8: Stopping monitors for $\hat{\alpha} = 10^1$ (left) and $\hat{\alpha} = 10^{-1}$ (right). x_c and x_f are the multiscale solution vector on the coarse and fine grids, respectively.

Next, we numerically investigate how the efficiency performance of i-ApMsFEM depends on the choices of the smoothers and the number of smoothing steps. Table 1 summarizes the results for the iteration counts. Two types of smoothers are considered, including the ν-step zero fill-in incomplete LU decomposition (ILU(0)) smoother and the ν-step incomplete LU with the threshold (ILUT(τ)) preconditioned GMRES method [24]. Here, the number of steps ν is varied from 1 to 3. Note that the threshold parameters τ is used to control the number of the fill-in elements in the upper and lower triangular decomposition. In the case of $\tau = 0$, ILUT is equivalent to the LU decomposition. On the other hand, when τ is set to be large enough, ILUT is reduced to ILU(0). In this set of the numerical experiment, τ is set to be 10^{-4} unless otherwise stated. From the table, we observe that

1. The i-ApMsFEM with ILU(0) as a smoother works fine for a small and moderate value of $\hat{\alpha}$. In general, the more smoothing steps are used, the fewer iteration counts for the i-ApMsFEM are needed. But for the high-contrast cases, it fails to converge when it reaches the maximum number of iterations, which is set to be 50 which but still does not meet the convergence criteria.
2. On the other hand, the ILUT-preconditioned GMRES method as a smoother is quite effective. Only a few iterations are needed, and the iteration counts are nearly independent of the values of $\hat{\alpha}$. Also, along with the results shown in Table 2 for both the double inclusions problem and the lens problem, the performance of i-ApMsFEM is excellent if an appropriate smoother is used for the different geometrical settings.

To understand why the convergence of the i-ApMsFEM with the ILUT preconditioned GMRES type smoother is so efficient, we plot the pointwise residual on the fine grid before and after performing the

ν	$\hat{\alpha}=10^1$	$\hat{\alpha}=10^3$	$\hat{\alpha}=10^5$	$\hat{\alpha}=10^{-1}$	$\hat{\alpha}=10^{-3}$	$\hat{\alpha}=10^{-5}$
ILU(0) smoother						
1	22	22	div.	26	26	div.
2	16	16	div.	19	19	div.
3	13	13	div.	16	15	div.
ILUT preconditioned GMRES smoother						
1	5	6	11	6	6	13
2	4	5	6	5	6	11
3	4	4	6	5	5	8

Table 1: A comparison of different types of smoothers and a varied numbers of smoothing steps for the single inclusion problem and corresponding the iteration counts.

ν	$\hat{\alpha}=10^1$	$\hat{\alpha}=10^3$	$\hat{\alpha}=10^5$	$\hat{\alpha}=10^{-1}$	$\hat{\alpha}=10^{-3}$	$\hat{\alpha}=10^{-5}$
Double inclusions problem						
1	5	5	13*	6	7	10
2	4	4	8*	5	6	7
3	4	4	8*	5	6	5
Lens problem						
1	5	5	10*	6	6	9
2	4	4	7*	5	6	9
3	4	3	5*	5	6	6

Table 2: A summary of the i-ApMsFEM iteration counts for the double inclusions problem and the lens problem. The symbol "*" indicates that more restricted threshold number $\tau = 10^{-5}$ is used for the ILUT GMRES smoother.

smoothing phase as shown in Figure 9. This figure indicates that before smoothing phase, the large pointwise residual always occurs near the edges of the coarse elements due to the error introduced by the inaccurate boundary condition settings. On the other hand, such high-frequency errors are soon removed after performing the two-step ILUT preconditioned GMRES iterations. Hence, the approximate solution of the immersed finite element method for the elliptic interface problem on the fine grid is found by five cycles of the i-ApMsFEM algorithm, which provides an accurate piece of information for setting the boundary condition of the local problem.

3.4. Convergence analysis of the i-ApMsFEM algorithm

Tables 3 and 4 show the L_2-norm and H^1-norm errors for the initial and final i-ApMsFEM solutions on the coarse grid sizes for different ratios, $\hat{\alpha}$, and their convergence rate. For the purpose of comparison, we also include the numerical results obtained by using the immersed finite element method on the fine grid and the L_2-norm error data available in Millward's thesis for an iterative multiscale finite element method. The convergence rate is numerically computed using linear regression in the least squares sense. As expected, both the final i-ApMsFEM solution and the immersed finite element solution achieve the second order and first order convergence rates in the L_2 and H^1 error norms, respectively. Only one exception is the case of $\hat{\alpha} = 10^1$ for the L_2 norm error; its rate of convergence is slightly degraded. Finally, the i-ApMsFEM solution on the coarse grid converges quadratically in the L_2 sense to the immersed finite element solution on the fine grid, i.e., $O(h^2)$, which is better than the results for Millward's iterative multiscale finite element at $O(H^2)$. Hence, the i-ApMsFEM algorithm can also be used as an iterative method for solving the linear system on the fine grid.

Next, we consider the lens problem with geometrical singularity at two corners of the lens by estimating the rate of the convergence for the i-ApMsFEM solution as $p = \log_2(\tilde{E}(H)/\tilde{E}(H/2))$, where $\tilde{E}(H) = \|u^H - u^{H/2}\|_\infty$ and $\tilde{E}(H/2) = \|u^{H/2} - u^{H/4}\|_\infty$ and summarize the results in Table 5. As shown in this table, the final i-ApMsFEM solution achieves asymptotically the quadratic convergence for the cases that $\hat{\alpha} > 1$ and the superlinear convergence for the cases that $\hat{\alpha} < 1$. Note that the accuracy of the i-ApMsFEM algorithm depends on the choice of the multiscale basis function problem solvers and the local mesh size

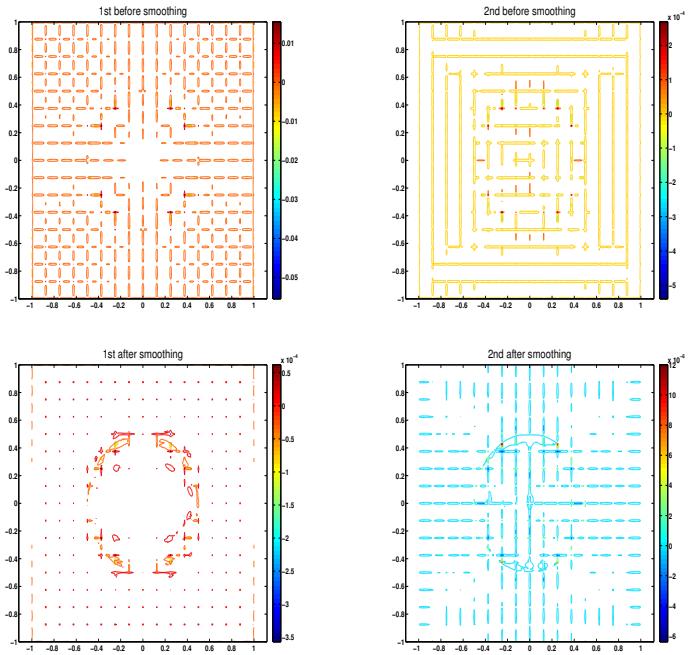

Figure 9: Pointwise residuals on the fine grid before (top row) and after (bottom row) smoothing phases at the first two iterations, $\hat{\alpha} = 10^{-1}$

is used. Hence the convergence rate of the method can be improved when more sophisticated numerical scheme [20] with finer local meshes is used.

4. Concluding remarks

In this work, we studied a new general framework of MsFEM that provides an accurate and effective numerical solution of PDE problems exhibiting the multiscale phenomena. The key ingredients of the MsFEM include a set of multiscale basis functions, which were constructed by solving locally the PDE problem with some appropriate boundary conditions and the coarse grid formulation, which take multiscale effects on the fine grid into accounts. The proposed algorithm, namely the iteratively adaptive multiscale finite element method algorithm, exchanges the local-global information by iteratively updating the local boundary condition for these multiscale basis functions. Then the multiscale solution provided feedback for updating the local boundary conditions on each coarse element. A few numerical results for the high contrast interface benchmark problems demonstrated the effectiveness of the proposed method. We believe that the i-ApMsFEM algorithm can be potentially applicable to other classes of multiscale problems, e.g., Helmholtz problems and is extendable to a system of multiscale PDE problems and nonlinear multiscale problems. Remarkably, the i-ApMsFEM algorithm can be viewed as the two-grid cycle multigrid type method and could be used as an efficient iterative solver for the fine-grid problems.

Acknowledgements

The authors were supported in part by the Ministry of Science and Technology of Taiwan, MOST-104-2115-M-008-011 and MOST-105-2115-M-008-007. The authors thank the anonymous reviewer and Prof. Thomas Y. Hou for constructive comments for improving the presentation of the manuscript.

L_2-error norm						
Initial i-ApMsFEM						
H	$\hat{\alpha}=10^{1}$	$\hat{\alpha}=10^{3}$	$\hat{\alpha}=10^{5}$	$\hat{\alpha}=10^{-1}$	$\hat{\alpha}=10^{-3}$	$\hat{\alpha}=10^{-5}$
1/2	2.3919E-2	1.0462E-2	1.1402E-2	2.0671E-1	2.3351E-1	2.5578E-1
1/4	1.4639E-2	3.6621E-2	3.7789E-2	5.5462E-2	5.9535E-2	5.9760E-2
1/8	9.1690E-3	3.3948E-2	3.6270E-2	1.8069E-2	2.6642E-2	2.6944E-2
1/16	4.3637E-3	1.7454E-2	2.1363E-2	6.5241E-3	1.4387E-2	1.4732E-2
rate	0.80	-0.21	-0.27	1.66	1.32	1.35
Final i-ApMsFEM						
1/2	2.6064E-4(3)	3.1145E-4(3)	7.7059E-4(3)	1.2476E-3(3)	1.2290E-3(3)	1.2770E-3(7)
1/4	6.7271E-5(3)	6.8259E-5(3)	1.0714E-4(3)	3.1426E-4(3)	3.0828E-4(5)	3.1433E-4(5)
1/8	1.9193E-5(3)	1.5899E-5(3)	3.1160E-5(4)	8.0581E-5(4)	7.7302E-5(5)	7.8847E-5(5)
1/16	7.7652E-6(3)	3.7600E-6(3)	8.4061E-6(5)	2.1565E-5(4)	1.9411E-5(4)	2.0016E-5(4)
rate	1.70	2.12	2.13	1.95	1.99	2.00
Immersed finite element method						
1/16	1.0570E-3	1.1630E-3	1.8528E-3	4.9343E-3	4.9200E-3	4.9929E-3
1/32	2.6064E-4	3.1145E-4	7.7059E-4	1.2476E-3	1.2290E-3	1.2769E-3
1/64	6.7271E-5	6.8259E-5	1.0709E-4	3.1426E-4	3.0828E-4	3.1433E-4
1/128	1.9193E-5	1.5899E-5	3.1159E-5	8.0581E-5	7.7302E-5	7.8826E-5
1/256	7.7653E-6	3.7600E-6	8.3792E-6	2.1558E-5	1.9411E-5	2.0017E-5
rate	1.79	2.08	2.02	1.96	2.00	1.99
Millward's iterative multiscale finite element method [23]						
1/4	1.0035E-2	7.7646E-3	7.8678E-3	6.9540E-2	6.8305E-2	6.7816E-2
1/8	2.9564E-3	3.2576E-3	3.0956E-3	1.7280E-2	1.7159E-2	1.6796E-2
1/16	8.4668E-4	7.8950E-4	8.0385E-4	4.3736E-3	4.3275E-3	4.1397E-3
1/32	2.2491E-4	2.0435E-4	2.0437E-4	1.0984E-3	1.0854E-3	1.0271E-3
1/64	5.8141E-5	5.2443E-5	5.2849E-5	2.7547E-4	2.7149E-4	2.6981E-4
rate	1.86	1.84	1.84	1.99	1.99	2.00

Table 3: The single inclusion case: A comparison of the L_2-error norm of the initial and final i-ApMsFEM solutions with $M = 16$, and the ones obtained by the immersed finite element method and Millward's method. The numbers in the parenthesis represents the number of the i-ApMsFEM iterations needed for convergence.

H^1-norm error						
Initial i-ApMsFEM						
H	$\hat{\alpha}=10^{1}$	$\hat{\alpha}=10^{3}$	$\hat{\alpha}=10^{5}$	$\hat{\alpha}=10^{-1}$	$\hat{\alpha}=10^{-3}$	$\hat{\alpha}=10^{-5}$
1/2	2.5639E-1	1.5941E-1	1.7177E-1	9.3010E-1	1.0330E+0	1.1353E+0
1/4	1.8845E-1	3.2388E-1	3.3251E-1	4.9111E-1	5.2452E-1	5.2676E-1
1/8	1.4989E-1	3.1968E-1	3.3729E-1	2.8072E-1	3.5393E-1	3.5642E-1
1/16	1.0344E-1	2.3043E-1	2.6505E-1	1.5972E-1	2.4536E-1	2.4927E-1
rate	0.43	-0.16	-0.19	0.84	0.68	0.71
Final i-ApMsFEM						
1/2	2.9871E-2(3)	2.9608E-2(3)	5.1684E-2(3)	1.0889E-1(3)	1.0891E-1(3)	1.1021E-1(7)
1/4	1.5076E-2(3)	1.1935E-2(3)	1.3207E-2(3)	5.4632E-2(3)	5.4429E-2(5)	5.4744E-2(5)
1/8	7.5261E-3(3)	5.8323E-3(3)	6.6288E-3(4)	2.7376E-2(4)	2.7191E-2(5)	2.7358E-2(5)
1/16	4.0591E-3(3)	2.8659E-3(3)	3.2433E-3(5)	1.3761E-2(4)	1.3576E-2(4)	1.3666E-2(4)
rate	0.96	1.11	1.30	0.99	1.00	1.00
Immersed finite element method						
1/16	5.8512E-2	5.0643E-2	6.0760E-2	2.1759E-1	2.1804E-1	2.2121E-1
1/32	2.9871E-2	2.9608E-2	5.1684E-2	1.0889E-1	1.0891E-1	1.1021E-1
1/64	1.5076E-2	1.1935E-2	1.3207E-2	5.4632E-2	5.4428E-2	5.4744E-2
1/128	7.5261E-3	5.8323E-3	6.6290E-3	2.7376E-2	2.7191E-2	2.7358E-2
1/256	4.0589E-3	2.8659E-3	3.2476E-3	1.3760E-2	1.3576E-2	1.3666E-2
rate	0.97	1.06	1.14	1.00	1.00	1.00

Table 4: The single inclusion case: A comparison of the H^1-error norm of the initial and final i-ApMsFEM solutions with $M = 16$ and the one that obtained by the immersed finite element method. The number in the parenthesis represents the number of the i-ApMsFEM iterations needed for convergence.

H	$\hat{\alpha}=10^{1}$	p	$\hat{\alpha}=10^{3}$	p	$\hat{\alpha}=10^{5}$	p
1/4	2.0669E-5		2.0659E-5		2.6936E-5	
1/8	7.0224E-6	1.56	1.0428E-5	0.99	1.0264E-5	1.39
1/16	1.9417E-6	1.85	2.1545E-6	2.28	2.5290E-6	2.02
H	$\hat{\alpha}=10^{-1}$	p	$\hat{\alpha}=10^{-3}$	p	$\hat{\alpha}=10^{-5}$	p
1/4	1.4231E-4		1.1542E-4		1.5833E-4	
1/8	5.2495E-5	1.44	4.4087E-5	1.39	1.0308E-4	0.62
1/16	1.8558E-5	1.50	1.3469E-5	1.71	3.9495E-5	1.38

Table 5: The lens problem. The convergence analysis for the final i-ApMsFEM solutions with $M = 16$.

References

[1] I.M. Babuska and S.A. Sauter. Is the pollution effect of the fem avoidable for the Helmholtz equation considering high wave numbers? *SIAM review*, 42:451–484, 2000.

[2] W.L. Briggs, V.E. Henson, and S.F. McCormick. *A Multigrid Tutorial*. SIAM, 2000.

[3] J.A. Burns, T. Lin, and L.G. Stanley. A Petrov Galerkin finite-element method for interface problems arising in sensitivity computations. *Comput. Math. Appl.*, 49:1889–1903, 2005.

[4] I.-L. Chern and Y.-C. Shu. A coupling interface method for elliptic interface problems. *J. Comput. Phys.*, 225:2138–2174, 2007.

[5] C.-C. Chu, Graham I.G., and T.-Y. Hou. A new multiscale finite element method for high-contrast elliptic interface problems. *Math. Comp.*, 79:1915–1955, 2010.

[6] R. Codina. A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation. *Comput. Methods Appl. Mech. Engrg.*, 110:325–342, 1993.

[7] Y. Efendiev, V. Ginting, T. Hou, and R. Ewing. Accurate multiscale finite element methods for two-phase flow simulations. *J. Comput. Phys.*, 220:155–174, 2006.

[8] Yalchin Efendiev and T.Y. Hou. *Multiscale Finite Element Methods. Theory and Applications*. Springer, 2009.

[9] L.P. Franca, C. Farhat, A.P. Macedo, and M. Lesoinne. Residual-free bubbles for the Helmholtz equation. *Int. J. Numer. Meth. Engrg. Eng.*, 40:4003–4009, 1997.

[10] L.P. Franca, S.L. Frey, and T.J.R. Hughes. Stabilized finite element methods: I. Application to the advective-diffusive model. *Computer Methods Appl. Mech. Engrg.*, 95:253–276, 1992.

[11] L.P. Franca and F.-N. Hwang. Refining the submesh strategy in the two-level finite element method: application to the advection–diffusion equation. *Inter. J. Numer. Meth. Fluids*, 39:161–187, 2002.

[12] L.P. Franca and A.P. Macedo. A two-level finite element method and its application to the Helmholtz equation. *Int. J Numer. Meth. Engrg.*, 43:23–32, 1998.

[13] L.P. Franca, A.L. Madureira, and F. Valentin. Towards multiscale functions: enriching finite element spaces with local but not bubble-like functions. *Comput. Methods Appl. Mech. Engrg.*, 194:3006–3021, 2005.

[14] H. Hajibeygi, G. Bonfigli, M.A. Hesse, and P. Jenny. Iterative multiscale finite-volume method. *J. Comput. Phys.*, 227:8604–8621, 2008.

[15] H. Hajibeygi and P. Jenny. Adaptive iterative multiscale finite volume method. *J. Comput. Phys.*, 230:628–643, 2011.

[16] X. He, T. Lin, and Y. Lin. Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions. *Int. J. Numer. Anal. Model*, 8:284–301, 2011.

[17] T.Y. Hou, F.-N. Hwang, P. Liu, and C.-C. Yao. An iteratively adaptive multi-scale finite element method for elliptic pdes with rough coefficients. *J. Comput. Phys.*, 336:375–400, 2017.

[18] T.Y. Hou and X.-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134:169–189, 1997.

[19] T.Y. Hou, X.-H. Wu, and Z. Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Math. Comput.*, 68:913–943, 1999.

[20] K. Li, Z. an Ito. *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*. SIAM, 2006.

[21] Z. Li. The immersed interface method using a finite element formulation. *Appl. Numer. Math.*, 27:253–267, 1998.

[22] Z. Li, T. Lin, and X. Wu. New Cartesian grid methods for interface problems using the finite element formulation. *Numer. Math*, 96:61–98, 2003.

[23] R.R. Millward. *A New Adaptive Multiscale Finite Element Method with Applications to High Contrast Interface Problems*. PhD thesis, University of Bath, 2011.

[24] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[25] Y.-T. Shih and H.C. Elman. Iterative methods for stabilized discrete convection-diffusion problems. *IMA J. Numer. Anal.*, 20:333–358, 2000.

[26] U. Trottenberg, C. Oosterlee, and A. Schűller. *Multigrid*. Academic Press, 2000.