

Multiscale eigenvalue computation for the flow equation and for the mixed boundary of diffusion model with small parameter

Shan Jiang

College of Mathematics Science, Yangzhou University

jiangshan@yzu.edu.cn

Supported by NSFC 11301462 and Jiangsu 13KJB110030

June 25, 2016

Outline

- 1 Introduction to multiscale computation
- 2 Mixed boundary of diffusion model with small parameter
- 3 Multiscale eigenvalue computation for flow equation

Outline

- 1 Introduction to multiscale computation
- 2 Mixed boundary of diffusion model with small parameter
- 3 Multiscale eigenvalue computation for flow equation

- **Multiscale computation** is a comprehensive topic, which has plenty of applications in scientific engineering. From either **macroscopical** or **microscopic** point of view, both have real disadvantages. In this way, multiscale computation may provide bridges among those complicated simulations.
- Let N be the number of elements in each spatial direction, and let M be the number of subcell elements, then there are a total of $O(M^d N^d)$ (d is the dimension) elements at the fine grid level in the **finite element method**, which leads to too extremely huge system to solve on fine grid size $h = 1/(MN)$.
- On the contrary, the **multiscale finite element method** only costs $O(M^d + N^d)$, and it solves on coarse grid size $H = 1/N$. As a consequence, it has great advantages and could be utilized in parallel computers.

- There are kinds of multiscale computations in recent research:

Multiscale Finite Element Method(MsFEM);

Generalized Multiscale Finite Element Method(GMsFEM);

Heterogeneous Multiscale Method;

Upscaling Method;

Homogenization Method;

Variational Multiscale Method;

Multiscale Finite Volume Method;

Multiscale Discontinuous Galerkin Method, etc..

Outline

- 1 Introduction to multiscale computation
- 2 Mixed boundary of diffusion model with small parameter
- 3 Multiscale eigenvalue computation for flow equation

- Consider the 1D convection-diffusion model

$$\begin{cases} Lu := -\varepsilon u''(x) + b_\varepsilon(x)u'(x) + c_\varepsilon(x)u(x) = f(x), & \text{in } x \in I = (0, 1), \\ k_1 u(0) + k_2 u'(0) = u_L, & k_3 u(1) + k_4 u'(1) = u_R, \end{cases} \quad (1)$$

where $u(x)$ is solution, $b_\varepsilon(x), c_\varepsilon(x)$ are rapidly oscillatory coefficients. And k_i are different constants to represent different kinds of boundary conditions, such as **Dirichlet, Neumann, or Robin types**.

- The weak form of (1) is to seek $u \in H^1$ such that

$$a(u, v) = (f, v), \quad \forall v \in H^1, \quad (2)$$

where the bilinear form is

$$a(u, v) = \int_0^1 (\varepsilon u' v' + b_\varepsilon(x) u' v + c_\varepsilon(x) uv) dx, \quad (3)$$

$$(f, v) = \int_0^1 f v dx. \quad (4)$$

- Besides from the **uniform** grid, we build several special grids to adapt the boundary layer information, which is according to a prior estimate. In the case of ε is small, we take the transition point $\tau = \min\{\frac{1}{2}, \frac{2\varepsilon}{\beta} \ln N\}$, and define the adapted grids with respect to the location of τ and integer λ .

$$\text{Shishkin: } x_i = \begin{cases} \frac{2\tau}{N} \cdot (i-1), & i = 1, \dots, \frac{N}{2} + 1, \\ \tau + \frac{2(1-\tau)}{N} \cdot (i - \frac{N}{2} - 1), & i = \frac{N}{2} + 2, \dots, N + 1. \end{cases} \quad (5)$$

$$\text{Graded: } x_i = \begin{cases} \frac{2(1-\tau)}{N} \cdot (i-1), & i = 1, \dots, \frac{N}{2} + 1, \\ 1 - \tau \left(\frac{2(N+1-i)}{N} \right)^\lambda, & i = \frac{N}{2} + 2, \dots, N + 1. \end{cases} \quad (6)$$

$$\text{Bakhvalov: } x_i = \begin{cases} -\varepsilon \ln \left[1 + \frac{4(1-N)(i-1)}{N^2} \right], & i = 1, \dots, \frac{N}{4} + 1, \\ \tau + \frac{2(1-2\tau)}{N} \cdot \left(i - \frac{N}{4} - 1 \right), & i = \frac{N}{4} + 2, \dots, \frac{3N}{4} + 1, \\ 1 + \varepsilon \ln \left[1 + \frac{4(1-N)(N+1-i)}{N^2} \right], & i = \frac{3N}{4} + 2, \dots, N + 1. \end{cases} \quad (7)$$

- In the boundary layers, we solve the local problem for the **multiscale basis functions** φ_i on each coarse element K ,

$$\begin{cases} L\varphi_i := -\varepsilon\varphi_i''(x) + b_\varepsilon(x)\varphi_i'(x) + c_\varepsilon(x)\varphi_i(x) = 0, & \text{in } K, \\ \varphi_i(x_j) = \delta_{ij}, & \text{on } \partial K. \end{cases} \quad (8)$$

- Since the original problem (1) and local problem (8) have the same differential operator, we solve it to obtain the boundary layer microscopic information through the multiscale basis functions automatically.
- As for the traditional finite element method, we take the **linear basis function** $\psi_i = 1 - \xi$, $\psi_{i+1} = \xi$, where $\xi = \frac{x-x_i}{h_i}$ and $h_i = x_{i+1} - x_i$.

$$\begin{aligned}
u_h &= u_i \psi_i + u_{i+1} \psi_{i+1} \\
&= u_i \frac{x_{i+1} - x}{x_{i+1} - x_i} + u_{i+1} \frac{x - x_i}{x_{i+1} - x_i}, \quad x \in I_i, \quad i = 1, 2, \dots, n. \\
u'_h &= u_i \psi'_i + u_{i+1} \psi'_{i+1} \\
&= u_i \frac{-1}{x_{i+1} - x_i} + u_{i+1} \frac{1}{x_{i+1} - x_i}, \quad x \in I_i, \quad i = 1, 2, \dots, n.
\end{aligned}$$

- If for the case of **Robin boundary condition** $u(0) = u_L, u(1) + u'(1) = u_R$,

$$\begin{cases} u_1 = u_L, \\ u_n \frac{-1}{h} + u_{n+1} \frac{1+h}{h} = u_R. \end{cases}$$

To define the first and last rows, and we assemble the global linear equations

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{-1}{h} & \frac{1+h}{h} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \\ u_{n+1} \end{pmatrix} = \begin{pmatrix} u_L \\ \vdots \\ \vdots \\ \vdots \\ u_R \end{pmatrix}.$$

Example 1. Model (1) with the mixed Robin boundary, whose exact solution is

$$u(x) = \frac{e^{m_1 x} - e^{m_2 x}}{(1 + m_1)e^{m_1} - (1 + m_2)e^{m_2}},$$

where $m_1 = \frac{-1 + \sqrt{1 + 4\varepsilon}}{2\varepsilon}$, $m_2 = \frac{-1 - \sqrt{1 + 4\varepsilon}}{2\varepsilon}$.

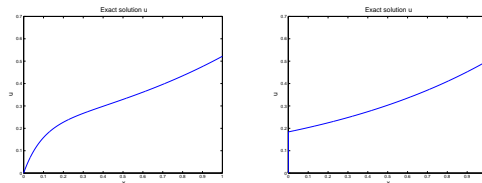


Fig. 1. Exact solution for cases of $\varepsilon = 10^{-1}$, 10^{-4} , respectively.

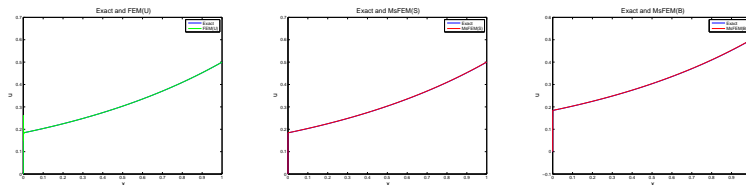


Fig. 2. When $\varepsilon = 10^{-4}$, exact solution and numerical solution of FEM(U) on $NM = 2048$ and MsFEM(S), MsFEM(B) on $N = 512$.

NM	FEM(U)	FEM(S)	FEM(B)	N	MsFEM(U)	MsFEM(S)	MsFEM(B)
16	1.793e-1	4.989e-2	6.804e-3	4	1.863e-1	4.287e-2	3.193e-2
32	2.394e-1	4.038e-2	3.348e-3	8	1.224e-1	1.919e-2	1.568e-2
64	1.679e-1	2.736e-2	1.736e-3	16	4.634e-2	8.465e-3	7.842e-3
128	3.439e-2	7.935e-3	9.475e-4	32	1.792e-2	3.800e-3	3.978e-3
256	4.425e-3	5.554e-4	4.905e-4	64	5.399e-3	2.041e-3	2.059e-3
512	1.178e-3	2.531e-4	2.451e-4	128	1.661e-3	1.064e-3	1.101e-3
1024	3.215e-4	1.268e-4	1.225e-4	256	5.623e-4	5.746e-4	6.291e-4
2048	9.120e-5	6.350e-5	6.125e-5	512	2.116e-4	3.332e-4	4.017e-4

Table 1. When $\varepsilon = 10^{-4}$, H^1 norm error of FEM(U), FEM(S), FEM(B) and MsFEM(U), MsFEM(S), MsFEM(B), respectively.

From the above figure and table, we can see when $\varepsilon = 10^{-4}$ is small to bring the boundary layer phenomena. The accuracy of traditional FEM on uniform grid level NM is bad, and it presents divergent on Graded grid. While the MsFEM just spends less computational costs on the coarse grid level N to acquire the accurate result, and with the mesh refinement its H^1 norm gives the first-order convergence.

Example 2. Model (1) with Dirichlet boundary and oscillatory coefficients

$$b_\varepsilon(x) = \left(\frac{1.8 + \sin\left(\frac{2\pi x}{\varepsilon_1}\right)}{1.8 + \cos\left(\frac{2\pi x}{\varepsilon_2}\right)} + \frac{1.8 + \cos\left(\frac{2\pi x}{\varepsilon_3}\right)}{1.8 + \sin\left(\frac{2\pi x}{\varepsilon_4}\right)} \right), c_\varepsilon(x) = \left(\frac{1.2 + \cos\left(\frac{2\pi x}{\varepsilon_4}\right)}{1.2 + \sin\left(\frac{2\pi x}{\varepsilon_3}\right)} + \frac{1.2 + \sin\left(\frac{2\pi x}{\varepsilon_2}\right)}{1.2 + \cos\left(\frac{2\pi x}{\varepsilon_1}\right)} \right).$$

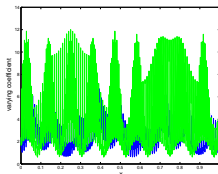
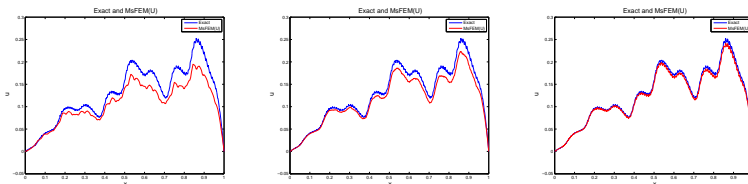


Fig. 3. Coefficient $b_\varepsilon(x)$ (blue), $c_\varepsilon(x)$ (green).

Take $\varepsilon = 0.1$, $\varepsilon_1 = 1/3$, $\varepsilon_2 = 1/9$, $\varepsilon_3 = 1/99$, $\varepsilon_4 = 1/200$, and $f(x) = 1$. Since there is no exact solution here, we use the FEM solution on very fine grid $NM = 8192$ as 'exact solution', and compare with the MsFEM solution on coarse grid N .

N	M	L^2 norm	abs error
32	8	0.1165	2.470e-2
64	8	0.1290	1.219e-2
128	8	0.1284	1.278e-2
256	8	0.1348	6.410e-3
512	8	0.1353	5.881e-3
1024	8	0.1365	4.684e-3

Table 2. L^2 norm and absolute error of MsFEM.Fig. 4. Exact solution and MsFEM solution (on $N = 32, 128, 1024$ grids), respectively.

From the figure and table, MsFEM solution approximates the reference solution accurately and efficiently with the mesh refinement.

Outline

- 1 Introduction to multiscale computation
- 2 Mixed boundary of diffusion model with small parameter
- 3 Multiscale eigenvalue computation for flow equation**

- Consider the flow equation

$$\begin{cases} \frac{\partial u}{\partial t} - \operatorname{div}(k(\mathbf{x})\nabla u) = f & \text{in } \Omega, \\ u = u_D & \text{on } \Gamma_D, \\ -k(\mathbf{x})\nabla u \cdot \mathbf{n} = f_N & \text{on } \Gamma_N, \end{cases} \quad (9)$$

where $k(\mathbf{x})$ is rapidly oscillatory coefficient, u_D , f_N is Dirichlet and Neumann boundary, respectively.

- The corresponding weak form is

$$a(u, v) = (f, v) - \langle f_N, v \rangle_{\Gamma_N}, \quad \forall v \in H_D^1, \quad (10)$$

where $a(u, v) = \int_{\Omega} k(\mathbf{x})\nabla u \cdot \nabla v d\mathbf{x}$, $(f, v) = \int_{\Omega} f v d\mathbf{x}$, $\langle f_N, v \rangle_{\Gamma_N} = \int_{\Gamma_N} f_N v ds$.

- To solve the local problem on coarse elements K to get the multiscale basis χ_i ,

$$\begin{cases} -\operatorname{div}(k(\mathbf{x})\nabla\chi_i) = 0 & \text{in } K, \\ \chi_i = \psi_i & \text{on } \partial K. \end{cases} \quad (11)$$

- Then every four coarse elements K form one coarse patch ω_i , and take $\tilde{k} = H^2 \sum_i k(\mathbf{x})\nabla\|\chi_i\|^2$. In this way, we apply the MsFEM enriched with the eigenvalue computation,

$$\begin{cases} -\operatorname{div}(k(\mathbf{x})\nabla\phi_i) = \lambda_i\tilde{k}\phi_i & \text{in } \omega_i, \\ -\nabla\phi_i \cdot \mathbf{n} = 0 & \text{on } \partial\omega_i, \end{cases} \quad (12)$$

we obtain the multiscale basis ϕ_i , and the ordering eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_L \leq \cdots$, where L is the chosen number of eigenvalues.

- Define $\Phi_{i,l} = \chi_i\phi_{i,l}$, where $1 \leq l \leq L$. We construct the multiscale space which be enriched with the eigenvalue pairs, such that

$$U^H = \operatorname{span}\{\Phi_{i,l} \subset H_D^1\}. \quad (13)$$

- For the multiscale basis functions to be enriched with eigenvalue information, we assemble the global mapping matrix according to the mesh nodes,

$$R = [\phi_1, \phi_2, \dots, \phi_L]. \quad (14)$$

The size of R would increase with respect to the number of L . But it still has great advantage than that of FEM on very fine grids.

- On coarse patch scale to solve the eigenvalue problem

$$A\Phi = \lambda M\Phi, \quad (15)$$

where $A = (A_{m,n}) = \int_{\omega_i} \tilde{k}(\mathbf{x}) \nabla \phi_n \cdot \nabla \phi_m = R^T \bar{A} R$,

$M = (M_{m,n}) = \int_{\omega_i} \tilde{k}(\mathbf{x}) \phi_n \cdot \phi_m = R^T \bar{M} R$.

For example, $L = 1$, $\text{size}(A) = 121^2$; $L = 2$, $\text{size}(A) = 202^2$;
 $L = 4$, $\text{size}(A) = 364^2$; $L = 8$, $\text{size}(A) = 688^2$. All of these are much smaller than that of FEM on very fine grids $(NM)^2 = 100^2$, whose $\text{size}(A) = 10201^2$. As a consequence, our enriched MsFEM can save plenty of computational resources.



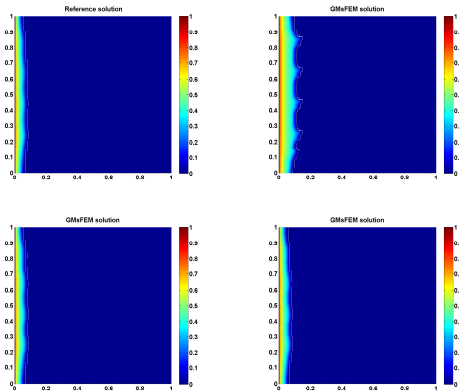


Fig. 6. At the same moment, **reference solution**, **enriched multiscale solution**
as for taking eigenvalue number of $L = 2, 4, 8$.

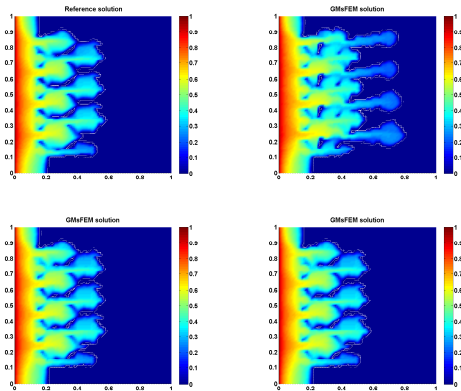


Fig. 7. At the same moment, **reference solution**, **enriched multiscale solution**
as for taking eigenvalue number of $L = 2, 4, 8$.

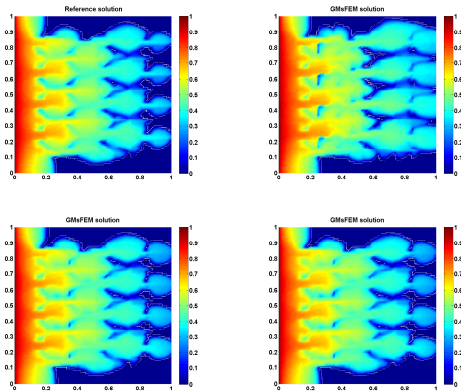


Fig. 8. At the same moment, **reference solution**, **enriched multiscale solution**
as for taking eigenvalue number of $L = 2, 4, 8$.

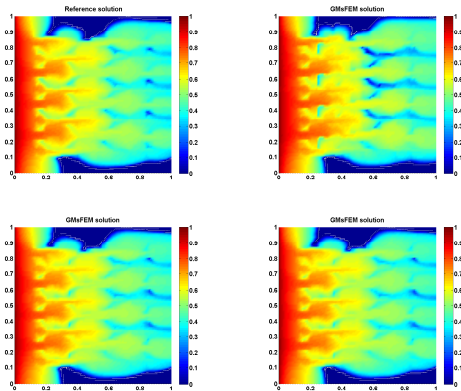


Fig. 9. At the same moment, **reference solution**, **enriched multiscale solution** as for taking eigenvalue number of $L = 2, 4, 8$.