Original Research Article

Adaptive Multilevel Monte Carlo Method for Elliptic Eigenvalue Problem with Random Coefficients

Abstract

For the elliptic eigenvalue problem with random coefficients, this paper establishes an adaptive multilevel Monte Carlo algorithm. This algorithm integrates the traditional multilevel Monte Carlo method with the adaptive finite element method, distributing samples across multiple levels. At each level, the corresponding deterministic eigenvalue problem for each sample is solved using the pathwise a posteriori error estimator and adaptive mesh refinement techniques to obtain an approximate solution that meets the preset accuracy. This sample-path-based mesh refinement strategy results in mutually independent hierarchical meshes for different samples. We provide the complexity analysis of the algorithm and demonstrate through a series of numerical experiments that the proposed algorithm can improve computational accuracy and reduce computational costs.

Keywords: multilevel Monte Carlo; adaptive mesh refinement; stochastic eigenvalue problem; complexity analysis

2020 Mathematics Subject Classification: 65N30; 65C05; 65N15

1 Introduction

In scientific computing and engineering applications, identifying, quantifying, and interpreting various types of uncertainties arising from mathematical models and input data have become increasingly important. The Monte Carlo (MC) method has been widely adopted for uncertainty quantification due to its robustness and simplicity. However, its convergence rate is relatively slow, and each sample requires solving a deterministic problem, leading to extremely high computational cost when high accuracy is required. To improve the efficiency of the traditional MC method, several improved techniques have been developed, such as Quasi-Monte Carlo (QMC) methods (Gilbert et al., 2019; Kuo et al., 2012) and Multilevel Monte Carlo (MLMC) methods (Charrier et al., 2013; Cliffe et al., 2011). As a variance-reduction technique, the MLMC method was initially proposed by Heinrich to approximate high-dimensional integrals and solve integral equations (Heinrich, 2001), and later extended by Giles to address integration problems related to stochastic differential equations (see (Giles, 2008, 2015)). This method combines the MC method with the multigrid approach by introducing a hierarchy of subproblems corresponding to mesh levels. Compared with the traditional MC method, the MLMC method significantly improves computational efficiency and has become a powerful tool

in various applications (Barth et al., 2011; Charrier et al., 2013; Cliffe et al., 2011; Teckentrup et al., 2013).

Although substantial progress has been made with the MLMC method, further improving its computational efficiency remains an active area of research. Existing studies can be generally categorized into two directions: one focuses on optimizing the given quasi-uniform mesh hierarchy, improving sample distribution strategies and parameter configurations to minimize overall computational costs (Collier et al., 2015; Haji-Ali et al., 2016); the other introduces adaptive mesh refinement techniques to dynamically adjust mesh resolution according to the local error features of the solution, thereby reducing computational costs while maintaining accuracy. There have been several exploratory works combining adaptive mesh refinement with MLMC (Detommaso et al., 2019; Eigel et al., 2016; Elfverson et al., 2016; Gerstner and Heinz, 2012; Hoel et al., 2011, 2014; Khodadadian et al., 2020), however, to the best of our knowledge, such studies on eigenvalue problems for partial differential equations have not yet been reported.

In this paper, we consider the following stochastic eigenvalue problem:

$$-\nabla \cdot (a(\mathbf{x},\omega)\nabla u(\mathbf{x},\omega)) + b(\mathbf{x},\omega)u(\mathbf{x},\omega) = \lambda(\omega)u(\mathbf{x},\omega), \quad \text{in } \mathbf{x} \in D, \tag{1.1}$$

$$u(\mathbf{x},\omega) = 0, \qquad \text{on } \mathbf{x} \in \Gamma := \partial D, \tag{1.2}$$

where $D \subset \mathbb{R}^2$ is a bounded polygonal domain, and ω denotes a random event defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The coefficients $a(\mathbf{x},\omega)$ and $b(\mathbf{x},\omega)$ play the role of sources of uncertainty in this problem. Therefore, the output eigenvalue $\lambda(\omega)$ and its corresponding eigenfunction $u(\mathbf{x},\omega)$ exhibit randomness and depend on the random parameter ω. Such problems are of great significance in various applications, including nuclear reactor physics (Duderstadt and Hamilton, 1976; Jamelot and Ciarlet Jr, 2013; Scheichl, 1997) and photonics (Dobson, 1999; Giani and Graham, 2012; Kuchment, 2001; Norton and Scheichl, 2013), as well as in quantum physics, acoustics, electromagnetic and elastic wave propagation, and structural mechanics.

In this paper, we focus on the expectation of the smallest eigenvalue of the problem (1.1)-(1.2):

$$\mathbb{E}[\lambda] = \int_{\Omega} \lambda(\omega)\, d\mathbb{P}(\omega).$$

In (Kornhuber and Youett, 2018; Youett, 2018), Kornhuber and Youett proposed adaptive MLMC finite element methods based on deterministic adaptive mesh refinement to solve "pathwise" problems for stochastic variational inequalities and stochastic elliptic equations, respectively. They established the corresponding convergence theories from the perspective of balancing accuracy and computational cost. By constructing a sequence of decreasing tolerances and employing adaptive meshes at each level consistent with the given tolerance, they achieved an optimal balance between accuracy and computational complexity. Based on the ideas in (Kornhuber and Youett, 2018; Youett, 2018), this paper develops an adaptive MLMC algorithm for the stochastic elliptic eigenvalue problem (1.1)-(1.2). The algorithm dynamically adjusts the mesh resolution at each level according to a prescribed tolerance sequence, progressively improving the accuracy of the approximate expectation. Specifically, the algorithm first sets initial parameters, including the number of samples and the tolerance; at each level, for a given sample, the path-adaptive finite element method is used to compute the approximate eigenvalues and eigenfunctions; then, the variance between adjacent levels is used to update the optimal number of samples; If the optimal number of samples exceeds the given sample size, additional samples are computed. If the convergence criteria are not yet satisfied, a new level is added. This process is repeated until the desired accuracy is achieved.

The remainder of this paper is organized as follows. In Section 2, we present the variational formulation of the stochastic eigenvalue problem (1.1)-(1.2) and its conforming finite element discretization, along with the a priori and a posteriori error estimates of the finite element solution. In Section 3, we establish an adaptive MLMC algorithm for the problem (1.1)-(1.2) and provide its complexity analysis. Then, in Section 4 we report some numerical experiments to demonstrate the effectiveness of the proposed algorithm and its advantages over the traditional MLMC method. Finally, in Section 5, we give the conclusion.

2 Finite Element Discretization and Error Estimation

In this section, let $\omega \in \Omega$ be fixed. For each $\omega \in \Omega$, we apply the standard finite element method to discretize the problem (1.1)-(1.2) in the spatial domain.

Following (Gilbert and Scheichl, 2024), we make the following assumptions on the coefficients $a(\mathbf{x}, \omega)$ and $b(\mathbf{x}, \omega)$.

Assumption 1.

- For all $(\mathbf{x}, \omega) \in D \times \Omega$, $a(\mathbf{x}, \omega)$ and $b(\mathbf{x}, \omega)$ belong to $L^\infty(D)$. For convenience, suppose there exists $a_{\max} < \infty$ such that

$$\max\left\{\sup_{\omega\in\Omega}\|a(\omega)\|_{L^\infty(D)}, \sup_{\omega\in\Omega}\|\nabla a(\omega)\|_{L^\infty(D)}, \sup_{\omega\in\Omega}\|b(\omega)\|_{L^\infty(D)}\right\} \le a_{\max}.$$

- There exists $a_{\min} > 0$ such that for all $\mathbf{x} \in D$ and $\omega \in \Omega$, it holds that $a(\mathbf{x}, \omega) \ge a_{\min}$ and $b(\mathbf{x}, \omega) \ge 0$.

For simplicity of notation, we sometimes omit the spatial coordinates of random functions and denote $a(\omega) := a(\cdot, \omega)$, $b(\omega) := b(\cdot, \omega)$, and $u(\omega) := u(\cdot, \omega)$.

Let $H^s(D)$ denote the usual Sobolev space of order s over D with norm $\|\cdot\|_{s,D}$, and $H^0(D) = L^2(D)$. Denote $V := H_0^1(D)$, equipped with the norm $\|v\|_{V,D} := \|\nabla v\|_{0,D}$, which is equivalent to $\|v\|_{1,D}$ on V.

The variational formulation of the eigenvalue problem (1.1)-(1.2) reads: find a nontrivial eigenpair $(\lambda(\omega), u(\omega)) \in \mathbb{R} \times V$ with $\|u(\omega)\|_{0,D} = 1$ such that

$$A(\omega; u(\omega), v) = \lambda(\omega) M(u(\omega), v), \quad \forall v \in V, \tag{2.1}$$

where

$$A(\omega; u(\omega), v) := \int_D a(\mathbf{x}, \omega)\nabla u(\mathbf{x}, \omega) \cdot \nabla v(\mathbf{x})\, d\mathbf{x} + \int_D b(\mathbf{x}, \omega) u(\mathbf{x}, \omega) v(\mathbf{x})\, d\mathbf{x},$$
$$M(u(\omega), v) := \int_D u(\mathbf{x}, \omega) v(\mathbf{x})\, d\mathbf{x}.$$

The bilinear form $A(\omega; \cdot, \cdot)$ is uniformly bounded and coercive with respect to ω, i.e.,

$$A(\omega; w, v) \le a_{\max}\|w\|_{V,D}\|v\|_{V,D}, \quad \forall w, v \in V,$$
$$A(\omega; v, v) \ge a_{\min}\|v\|_{V,D}^2, \quad \forall v \in V,$$

where $a_{\min}$ and $a_{\max}$ are given in **Assumption 1**.

The variational eigenvalue problem (2.1) is self-adjoint, therefore (2.1) has a countable sequence of strictly positive eigenvalues (see (Babuška and Osborn, 1991)), which can be arranged in ascending order and counted with multiplicities as

$$0 < \lambda^1(\omega) \le \lambda^2(\omega) \le \lambda^3(\omega) \le \cdots \nearrow \infty,$$

with the corresponding eigenfunctions

$$u^1(\omega), u^2(\omega), u^3(\omega), \cdots.$$

Assumption 2 There exist a constant $C_{re} > 0$ and a parameter $s \in [0, 1]$ such that: for any $f \in L^2(D)$, if $w \in V$ solves the variational equation $A(\omega; w, v) = M(f, v), \forall v \in V$, then $\|w\|_{1+s,D} \le C_{re}\|f\|_{0,D}$.

When the coefficient $a(\omega)$ is a constant or a smooth function and the domain D has a smooth boundary or is a convex polygon, $s = 1$; when D is nonconvex (see (Bourlard et al., 1992)) or the coefficient $a(\omega)$ is discontinuous (see (Babuška, 1970)), then $s \in (0, 1)$. Under this assumption, the

eigenfunctions $u^j(\omega)$ of the problem (2.1) belong to $H^{1+s}(D)$ and satisfy $\|u^j(\omega)\|_{1+s,D} \leq C_{re}\lambda^j$.

Let $\{\mathcal{T}_h\}_{h>0}$ be a family of shape-regular triangulations of the spatial domain D, where the mesh diameter $h = \max\{\text{diam}(T) : T \in \mathcal{T}_h\}$. Let $V_h \subset V$ be the linear finite element space defined over $\mathcal{T}_h$, i.e.,

$$V_h := \{v \in H_0^1(D) : v|_T \in P_1(T), \forall T \in \mathcal{T}_h\},$$

where $P_1(T)$ denotes the space of polynomials of degree at most 1 on element T. The dimension of the finite element space V_h is $M_h := \dim(V_h) \approx h^{-2}$, which holds for quasi-uniform meshes or locally refined meshes.

For each point ω in the parameter space Ω, the finite element discretization of problem (2.1) reads: find $\lambda_h(\omega) \in \mathbb{R}$ and $u_h(\omega) \in V_h$ with $\|u_h(\omega)\|_{0,D} = 1$ such that

$$A(\omega; u_h(\omega), v_h) = \lambda_h(\omega)M(u_h(\omega), v_h), \quad \forall v_h \in V_h. \tag{2.2}$$

Problem (2.2) has a finite number of eigenvalues:

$$0 < \lambda_h^1(\omega) \leq \lambda_h^2(\omega) \leq \cdots \leq \lambda_h^{M_h}(\omega),$$

with the corresponding eigenfunctions denoted by

$$u_h^1(\omega), u_h^2(\omega), \ldots, u_h^{M_h}(\omega).$$

The following a priori error estimates for the approximate eigenvalues and eigenfunctions can be found in (Babuška and Osborn, 1991): let $\lambda^j(\omega)$ and $\lambda_h^j(\omega)$ be the jth eigenvalues of problems (2.1) and (2.2), respectively, and let $u_h^j(\omega)$ be the eigenfunction corresponding to $\lambda_h^j(\omega)$, then there exists an eigenfunction $u^j(\omega)$ of (2.1) such that

$$\left\|u_h^j(\omega) - u^j(\omega)\right\|_{V,D} \leq Ch^s, \tag{2.3}$$

$$\left\|u_h^j(\omega) - u^j(\omega)\right\|_{0,D} \leq Ch^s \left\|u_h^j(\omega) - u^j(\omega)\right\|_{V,D}, \tag{2.4}$$

$$|\lambda_h^j(\omega) - \lambda^j(\omega)| \leq C \left\|u_h^j(\omega) - u^j(\omega)\right\|_{V,D}^2, \tag{2.5}$$

where and hereafter C denotes a positive constant independent of the mesh size h, depending only on the specific eigenvalue and the regularity of $\mathcal{T}_h$.

Next we present a residual-type a posteriori error estimator for the finite element discretization (2.2). We begin by introducing some notation. Let $\mathcal{E}(T)$ denote the set of edges of an element $T \in \mathcal{T}_h$, and let $\mathcal{E}$ denote the set of all interior edges in the mesh $\mathcal{T}_h$. Let n_T denote the unit outward normal on the boundary ∂T of element $T \in \mathcal{T}_h$. For each edge $E \in \mathcal{E}$, let the two elements sharing this edge be denoted as T_{in} and T_{out}. Let n_E be the unit normal vector on E from T_{out} towards to T_{in}. For $v_h \in V_h$, we define the jump of the normal derivative of v_h across edge E by

$$J_E(v_h) = \nabla\Big(v_h|_{T_{out}}\Big) \cdot n_E - \nabla\Big(v_h|_{T_{in}}\Big) \cdot n_E.$$

Let $u(\omega)$ and $u_h(\omega)$ be the exact and finite element solutions of (2.1) and (2.2), respectively, and denote $e = u(\omega) - u_h(\omega)$. For any $v \in V$, applying Green's formula on each element yields

$$\begin{aligned}
&A(\omega; e, v) = A(\omega; u, v) - A(\omega; u_h, v) = M(\lambda u, v) - A(\omega; u_h, v) \\
&= \sum_{T\in\mathcal{T}_h} \int_T \lambda(\omega)u(\omega)v\,d\mathbf{x} - \sum_{T\in\mathcal{T}_h} \Big(\int_T (-\nabla\cdot(a(\omega)\nabla u_h(\omega)) + b(\omega)u_h(\omega))v\,d\mathbf{x} + \int_{\partial T} a(\omega)\nabla u_h(\omega)\cdot n_T v\,ds\Big) \\
&= \sum_{T\in\mathcal{T}_h} \int_T \big(\lambda(\omega)u(\omega) - \lambda_h(\omega)u_h(\omega)\big)v\,d\mathbf{x} \\
&\quad - \sum_{T\in\mathcal{T}_h} \int_T \big(-\nabla\cdot(a(\omega)\nabla u_h(\omega)) + b(\omega)u_h(\omega) - \lambda_h(\omega)u_h(\omega)\big)v\,d\mathbf{x} - \sum_{E\in\mathcal{E}} \int_E a(\omega)J_E(u_h)v\,ds,
\end{aligned} \tag{2.6}$$

and using $\|u(\omega)\|_{0,D}=1$ and $\|u_h(\omega)\|_{0,D}=1$, one obtains

$$\begin{aligned}
&\int_D (\lambda(\omega)u(\omega)-\lambda_h(\omega)u_h(\omega))(u(\omega)-u_h(\omega))\,d\mathbf{x}\\
&=\int_D \lambda(\omega)u^2(\omega)\,d\mathbf{x}-\int_D(\lambda(\omega)+\lambda_h(\omega))u(\omega)u_h(\omega)\,d\mathbf{x}+\int_D\lambda_h(\omega)u_h^2(\omega)\,d\mathbf{x}\\
&=(\lambda(\omega)+\lambda_h(\omega))\Big(1-\int_D u(\omega)u_h(\omega)\,d\mathbf{x}\Big)\\
&=\frac{\lambda(\omega)+\lambda_h(\omega)}{2}\int_D e^2\,d\mathbf{x}. \qquad (2.7)
\end{aligned}$$

For any $v\in V$, let $I_hv\in V_h$ denote the Clément interpolant of v (see (Clément, 1975)), then

$$\|v-I_hv\|_{0,T}\le Ch_T\|v\|_{V,\tilde{T}},\quad \|v-I_hv\|_{0,E}\le Ch_E^{\frac{1}{2}}\|v\|_{V,\tilde{E}},$$

where $h_T=\mathrm{diam}(T)$ is the diameter of T, h_E the diameter of edge E, $\tilde{T}$ the union of all elements sharing at least one vertex with T, and $\tilde{E}$ the union of all elements sharing at least one vertex with E.

Using (2.1) and (2.2), together with (2.6), (2.7), the Schwarz inequality, and the Clément interpolation estimates, we derive

$$\begin{aligned}
&A(\omega;e,e)=A(\omega;e,e-I_he)+A(\omega;e,I_he)\\
&=A(\omega;e,e-I_he)+M(\lambda(\omega)u(\omega)-\lambda_h(\omega)u_h(\omega),I_he)\\
&=\sum_{T\in\mathcal{T}_h}\int_T(\lambda(\omega)u(\omega)-\lambda_h(\omega)u_h(\omega))e\,d\mathbf{x}\\
&\quad-\sum_{T\in\mathcal{T}_h}\int_T\big(-\nabla\cdot(a(\omega)\nabla u_h(\omega))+b(\omega)u_h(\omega)-\lambda_h(\omega)u_h(\omega)\big)(e-I_he)\,d\mathbf{x}-\sum_{E\in\mathcal{E}}\int_E a(\omega)J_E(u_h)(e-I_he)\,ds\\
&\le\frac{\lambda(\omega)+\lambda_h(\omega)}{2}\|e\|_{0,D}^2+\sum_{T\in\mathcal{T}_h}\|\nabla\cdot(a(\omega)\nabla u_h(\omega))-b(\omega)u_h(\omega)+\lambda_h(\omega)u_h(\omega)\|_{0,T}\|e-I_he\|_{0,T}\\
&\quad+\sum_{E\in\mathcal{E}}\|a(\omega)J_E(u_h)\|_{0,E}\|e-I_he\|_{0,E}\\
&\le\frac{\lambda(\omega)+\lambda_h(\omega)}{2}\|e\|_{0,D}^2+C\sum_{T\in\mathcal{T}_h}\|\nabla\cdot(a(\omega)\nabla u_h(\omega))-b(\omega)u_h(\omega)+\lambda_h(\omega)u_h(\omega)\|_{0,T}h_T\|e\|_{V,\tilde{T}}\\
&\quad+C\sum_{E\in\mathcal{E}}\|a(\omega)J_E(u_h)\|_{0,E}h_E^{\frac{1}{2}}\|e\|_{V,\tilde{E}}. \qquad (2.8)
\end{aligned}$$

Define the local error indicator

$$\eta_T^2(\omega,\lambda_hu_h)=\|\nabla\cdot(a(\omega)\nabla u_h(\omega))-b(\omega)u_h(\omega)+\lambda_h(\omega)u_h(\omega)\|_{0,T}^2h_T^2+\frac{1}{2}\sum_{E\in\mathcal{E}(T)\cap\mathcal{E}}\|a(\omega)J_E(u_h(\omega))\|_{0,E}^2h_E,$$

and the global error indicator

$$\eta_{\mathcal{T}}(\omega,\lambda_hu_h)=\left(\sum_{T\in\mathcal{T}_h}\eta_T^2(\omega,\lambda_hu_h)\right)^{\frac{1}{2}},$$

then by the coercivity of the bilinear form $A(\omega;\cdot,\cdot)$, together with (2.4) and (2.8), we obtain the following reliability result of the error indicator.

Theorem 2.1. *For all $\omega \in \Omega$, let $(\lambda(\omega), u(\omega))$ be an eigenpair of* (2.1)*, and $(\lambda_h(\omega), u_h(\omega))$ its finite element approximation. Then there exists a constant C, depending only on the shape regularity of $\mathcal{T}_h$, such that*

$$\begin{aligned} \|u(\omega) - u_h(\omega)\|_{V,D} &\leq C\eta_{\mathcal{T}}(\omega, \lambda_h u_h), \\ |\lambda(\omega) - \lambda_h(\omega)| &\leq C\eta^2_{\mathcal{T}}(\omega, \lambda_h u_h). \end{aligned}$$

Using the bubble function technique (see Lemma 1.3 in (Verfürth, 1996)), the local efficiency of the indicator $\eta_{\mathcal{T}}(\omega, \lambda_h u_h)$ can be established. For $T \in \mathcal{T}_h$, let $\varphi_T \in H^1_0(T)$ denote the standard cubic bubble function taking the value 1 at the barycenter of T. By arguments similar to those used in proving standard local inverse inequalities (see Theorem 3.3 in (Ainsworth and Oden, 1997)), one can show the existence of a constant C depending only on the shape regularity of T, such that for $v_h \in P_1(T)$,

$$\|v_h\varphi_T\|_{0,T} \leq \|v_h\|_{0,T} \leq C\Big(\int_T |v_h|^2\varphi_T\Big)^{\frac{1}{2}}, \tag{2.9}$$

$$\|v_h\varphi_T\|_{V,T} \leq Ch_T^{-1}\|v_h\|_{0,T}. \tag{2.10}$$

Denote $R := R(\omega; \lambda_h(\omega), u_h(\omega)) = \nabla\cdot(a(\omega)\nabla u_h(\omega)) - b(\omega)u_h(\omega) + \lambda_h(\omega)u_h(\omega)$, and let $\overline{R} = \frac{1}{|T|}\int_T R(\omega; \lambda_h(\omega), u_h(\omega))\, d\mathbf{x}$ be the L^2 projection of R onto the space of piecewise constants. By the triangle inequality, it holds that

$$\|R\|_{0,T} \leq \|\overline{R}\|_{0,T} + \|R - \overline{R}\|_{0,T}.$$

Using (2.9) we have

$$\|\overline{R}\|^2_{0,T} \leq C\int_T \overline{R}^2\varphi_T\, d\mathbf{x} = C\int_T \overline{R}(\overline{R}\varphi_T)\, d\mathbf{x} = C\Big(\int_T R(\overline{R}\varphi_T)\, d\mathbf{x} + \int_T (\overline{R} - R)\overline{R}\varphi_T\, d\mathbf{x}\Big).$$

Let $v = \overline{R}\varphi_T$, and apply Green's formula on each element, noting that $v \in H^1_0(T)$ as well as the fact that for the eigenpair $(\lambda(\omega), u(\omega))$ of (2.1), it holds $\int_T (a(\omega)\nabla u(\omega)\cdot\nabla v + b(\omega)u(\omega)v - \lambda(\omega)u(\omega)v)\, d\mathbf{x} = 0$, thus we derive

$$\begin{aligned} &\int_T Rv\, d\mathbf{x} = \int_T (\nabla\cdot(a(\omega)\nabla u_h(\omega)) - b(\omega)u_h(\omega) + \lambda_h(\omega)u_h(\omega))\, v\, d\mathbf{x} \\ &= -\int_T \Big(a(\omega)\nabla u_h(\omega)\cdot\nabla v + b(\omega)u_h v - \lambda_h(\omega)u_h(\omega)v\Big)d\mathbf{x} + \int_T \Big(a(\omega)\nabla u(\omega)\cdot\nabla v + b(\omega)u(\omega)v - \lambda(\omega)u(\omega)v\Big)d\mathbf{x} \\ &= \int_T a(\omega)\nabla(u(\omega) - u_h(\omega))\cdot\nabla v + \int_T b(\omega)(u(\omega) - u_h(\omega))v - \int_T (\lambda(\omega)u(\omega) - \lambda_h(\omega)u_h(\omega)) \\ &\leq C\left(\|u(\omega) - u_h(\omega)\|_{V,T}\cdot\|v\|_{V,T} + \|\lambda(\omega)u(\omega) - \lambda_h(\omega)u_h(\omega)\|_{0,T}\cdot\|v\|_{0,T}\right), \end{aligned}$$

then by (2.9), (2.10) and the Schwarz inequality we get

$$\|\overline{R}\|^2_{0,T} \leq C\|u(\omega) - u_h(\omega)\|_{V,T}\cdot h_T^{-1}\|\overline{R}\|_{0,T} + \|\lambda(\omega)u(\omega) - \lambda_h(\omega)u_h(\omega)\|_{0,T}\cdot\|\overline{R}\|_{0,T} + \|\overline{R} - R\|_{0,T}\cdot\|\overline{R}\|_{0,T},$$

which implies

$$h_T\|\overline{R}\|_{0,T} \leq C\left(\|u(\omega) - u_h(\omega)\|_{V,T} + h_T\|\lambda(\omega)u(\omega) - \lambda_h(\omega)u_h(\omega)\|_{0,T} + h_T\|\overline{R} - R\|_{0,T}\right),$$

thus

$$h_T\|R\|_{0,T} \leq C\left(\|u(\omega) - u_h(\omega)\|_{V,T} + h_T\|\lambda(\omega)u(\omega) - \lambda_h(\omega)u_h(\omega)\|_{0,T} + h_T\|\overline{R} - R\|_{0,T}\right). \tag{2.11}$$

On the other hand, Lemma 3.4 in (Durán et al., 2003) shows that

$$h_E^{\frac{1}{2}}\|J_E(u_h)\|_{0,E} \leqslant C\Big(\|u(\omega) - u_h(\omega)\|_{V,T'} + h_{T'}\|\lambda(\omega)u(\omega) - \lambda_h(\omega)u_h(\omega)\|_{0,T'}\Big), \tag{2.12}$$

where $T' = T_{in} \cup T_{out}$.

Combining (2.11), (2.12) with the interpolation error estimates yields the following local efficiency result of the error indicator.

Theorem 2.2. *For $T \in \mathcal{T}_h$, let T^* be the union of elements sharing at least one edge with T, and $\widetilde{T}$ be the union of elements sharing at least one vertex with T. There exists a constant C depending only on the shape-regularity of the elements in T^* such that*

$$\eta_T(\omega, \lambda_h u_h) \leqslant C\Big(\|u(\omega) - u_h(\omega)\|_{V,T^*} + h_T\|\lambda(\omega)u(\omega) - \lambda_h(\omega)u_h(\omega)\|_{0,T^*} + h_T\|u_h\|_{1,\widetilde{T}}\Big).$$

3 Adaptive Multilevel Monte Carlo Method

In this section, we will introduce the adaptive mesh refinement strategy in the MLMC method and develop an adaptive multilevel Monte Carlo (Adaptive MLMC) algorithm for solving the problem (2.1). From this section onward, we focus solely on the expectation of the smallest eigenvalue of the stochastic elliptic problem, denoted by $\mathbb{E}[\lambda(\omega)]$. For simplicity, the notation $\lambda^1(\omega)$ is abbreviated as $\lambda(\omega)$.

We first recall the estimator of $\mathbb{E}[\lambda(\omega)]$ in the standard MC method, which takes the form

$$\mathbb{E}[\lambda(\omega)] \approx E_N[\lambda_h(\omega)] := \frac{1}{N}\sum_{i=1}^{N} \lambda_h(\omega_i), \tag{3.1}$$

where ω_i, $i = 1, \ldots, N$, are $N \in \mathbb{N}$ independent and identically distributed (i.i.d.) samples.

3.1 Multilevel Monte Carlo Method

Unlike the standard MC method, the MLMC method introduces a hierarchical structure with different discretization levels to achieve a balance between computational efficiency and accuracy. Given the initial mesh $\mathcal{T}_{h_0}$ with mesh diameter h_0, the MLMC method usually adopts a geometric refinement strategy: $h_\ell = h_0 * 2^{-\ell}$, with uniform refinement to generate meshes at level ℓ ($\ell \geq 1$), while the number of levels L is determined by the required accuracy or pre-set before computation.

Let $\lambda_{h_\ell}(\omega)$ denote the approximation of the smallest eigenvalue computed on the mesh $\mathcal{T}_{h_\ell}$ ($\ell = 0, \ldots, L$), and we write $\lambda_\ell(\omega) := \lambda_{h_\ell}(\omega)$ for simplicity. From (2.3) and (2.5), it can be seen that as the mesh is refined, the accuracy of approximate eigenvalues gradually improves, but the computational cost also increases accordingly. If one directly employs (3.1) to compute the approximate expectation, the required sample size N to meet the desired accuracy is large, resulting in high computational costs. To alleviate this, the MLMC method introduces the following telescoping decomposition:

$$\begin{aligned}\mathbb{E}[\lambda(\omega)] \approx \mathbb{E}[\lambda_L(\omega)] &= \sum_{\ell=0}^{L} \mathbb{E}[Y_\ell] \\ &= \mathbb{E}[\lambda_0(\omega)] + \sum_{\ell=1}^{L} \mathbb{E}[\lambda_\ell(\omega) - \lambda_{\ell-1}(\omega)],\end{aligned}$$

where $Y_0 := \lambda_0(\omega)$ and $Y_\ell := \lambda_\ell(\omega) - \lambda_{\ell-1}(\omega)$ for $\ell \geq 1$. The corresponding stochastic estimator can be written as

$$\mathbb{E}[\lambda_L(\omega)] \approx \frac{1}{N_0}\sum_{i=1}^{N_0} \lambda_0(\omega_i) + \sum_{\ell=1}^{L} \frac{1}{N_\ell}\sum_{i=1}^{N_\ell} \big(\lambda_\ell(\omega_i) - \lambda_{\ell-1}(\omega_i)\big), \tag{3.2}$$

where N_ℓ denotes the number of samples at level ℓ. Accordingly, the MLMC estimator reads

$$\hat{Y} := \sum_{\ell=0}^{L} \hat{Y}_\ell, \quad \hat{Y}_\ell := \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \big(\lambda_\ell(\omega_i) - \lambda_{\ell-1}(\omega_i)\big), \quad \hat{Y}_0 := \frac{1}{N_0} \sum_{i=1}^{N_0} \lambda_0(\omega_i), \tag{3.3}$$

and the mean squared error of the estimator $\hat{Y}$ is given by

$$\mathsf{MSE}\big(\mathbb{E}[\lambda(\omega)], \hat{Y}\big) = \sum_{\ell=0}^{L} \frac{\mathbb{V}[Y_\ell]}{N_\ell} + \left(\mathbb{E}[\lambda_L(\omega) - \lambda(\omega)]\right)^2.$$

To ensure that the total error meets the accuracy requirement: $\mathsf{MSE}(\mathbb{E}[\lambda(\omega)], \hat{Y}) \leq Tol^2$, we require the following two constraints to be satisfied simultaneously:

$$\mathbb{E}[\lambda_L(\omega) - \lambda(\omega)] \leq \frac{Tol}{\sqrt{2}}, \tag{3.4}$$

$$\sum_{\ell=0}^{L} \frac{\mathbb{V}[Y_\ell]}{N_\ell} \leq \frac{Tol^2}{2}. \tag{3.5}$$

Since the exact eigenvalue $\lambda(\omega)$ is unknown, (3.4) cannot be verified directly. To address this, we introduce an initial tolerance $Tol_0 > 0$ and a reduction factor $0 < q < 1$, and define a decreasing tolerance sequence:

$$Tol_\ell := q\, Tol_{\ell-1}, \quad \text{requiring} \quad |\lambda_\ell(\omega) - \lambda_{\ell-1}(\omega)| \leq Tol_\ell. \tag{3.6}$$

Under these conditions, following (Giles, 2015), we have

$$\mathbb{E}[\lambda_L(\omega) - \lambda(\omega)] = \sum_{\ell=L+1}^{\infty} \mathbb{E}[\lambda_\ell(\omega) - \lambda_{\ell-1}(\omega)] \leq (q^{-1}-1)^{-1} \mathbb{E}[\lambda_L(\omega) - \lambda_{L-1}(\omega)] \leq (q^{-1}-1)^{-1} Tol_L,$$

which allows us to replace (3.4) with

$$\mathbb{E}[\lambda_L(\omega) - \lambda_{L-1}(\omega)] = \frac{1}{N_L} \sum_{i=1}^{N_L} \big(\lambda_L(\omega_i) - \lambda_{L-1}(\omega_i)\big) \leq Tol_L \leq \left(q^{-1}-1\right) \frac{Tol}{\sqrt{2}}. \tag{3.7}$$

Next, to satisfy (3.5) and allocate the sample size appropriately, we introduce the following assumption:

Assumption 3. Let the computational cost of the approximate eigenvalue $\lambda_\ell(\omega)$ be C_ℓ. There exist constants $\gamma > 0$ and $C_{\mathsf{cost}} > 0$ such that

$$C_\ell \leqslant C_{\mathsf{cost}}\, M_{h_\ell}^{\gamma},$$

where M_{h_ℓ} denotes the number of degrees of freedom, which is independent of ℓ and ω.

Under this assumption, following (Youett, 2018), the following optimization problem can be formulated to minimize the total computational cost:

$$\min_{N_\ell} \sum_{\ell=0}^{L} \sum_{i=1}^{N_\ell} C_\ell(\omega_i), \quad \text{s.t.} \quad \sum_{\ell=0}^{L} \frac{\mathbb{V}[Y_\ell]}{N_\ell} = \frac{Tol^2}{2}.$$

Solving the above optimization problem gives the optimal sample size:

$$N_\ell^{\mathsf{opt}} = 2\, Tol^{-2} \sqrt{\frac{\mathbb{V}[Y_\ell]}{\sum_{i=1}^{N_\ell} C_\ell(\omega_i)}} \sum_{j=0}^{L} \sqrt{\mathbb{V}[Y_j] \sum_{i=1}^{N_\ell} C_j(\omega_i)}. \tag{3.8}$$

Hence, the MLMC method achieves optimal control of computational cost while ensuring the desired accuracy. The complete procedure of the algorithm is summarized as **Algorithm 1**.

Algorithm 1 Multilevel Monte Carlo Algorithm

Input: initial mesh $\mathcal{T}_{(0)}$, tolerance Tol, reduction factor q, and initial sample size N_{init}.
Initialize $L := 1$, set $N_\ell := N_{\text{init}}$ for $\ell = 0, 1$.
Generate N_0 samples and solve (2.2) on mesh $\mathcal{T}_{(0)}$ to obtain $\lambda_0(\omega_i)$, $i = 1, \ldots, N_0$.
Refine $\mathcal{T}_{(0)}$ to obtain $\mathcal{T}_{(1)}$, generate N_1 samples and solve (2.2) on $\mathcal{T}_{(1)}$ to obtain $\lambda_1(\omega_i)$, $i = 1, \ldots, N_1$.
while true **do**
    while $L > 1$ **do**
        Refine $\mathcal{T}_{(L-1)}$ to obtain $\mathcal{T}_{(L)}$, generate N_L samples and solve (2.2) on meshes $\mathcal{T}_{(L-1)}$ and $\mathcal{T}_{(L)}$ to obtain $\lambda_{L-1}(\omega_i)$ and $\lambda_L(\omega_i)$, $i = 1, \ldots, N_L$, respectively.
    end while
    Compute/update $\mathbb{V}[\lambda_\ell(\omega) - \lambda_{\ell-1}(\omega)]$ and optimal sample sizes N_ℓ^{opt} via (3.8), with $\lambda_{-1}(\omega) = 0$.
    if $N_\ell^{\text{opt}} > N_\ell$ **then**
        Generate additional $N_\ell^{\text{opt}} - N_\ell$ samples on levels $\ell = 0, \ldots, L$.
        Solve equation (2.2) on meshes $\mathcal{T}_{(\ell)}$ and $\mathcal{T}_{(\ell-1)}$ to obtain $\lambda_\ell(\omega_i)$ and $\lambda_{\ell-1}(\omega_i)$, respectively, where $\ell = 1, \ldots, L$, $i = 1, \ldots, N_\ell^{\text{opt}} - N_\ell$; for $\ell = 0$, solve equation (2.2) on $\mathcal{T}_{(0)}$.
    end if
    Update $N_\ell := N_\ell^{\text{opt}}$ for all levels.
    if convergence criterion (3.7) is satisfied **then**
        break
    else
        Increase the level: $L := L + 1$
    end if
end while
Compute the estimator $\mathbb{E}[\lambda_L]$ using (3.2).
Output: $\mathbb{E}[\lambda_L]$.

3.2 Adaptive Multilevel Monte Carlo Method

The standard MLMC method distributes samples across multiple levels and constructs an estimator of the expectation in a telescoping manner, thereby reducing the computational effort while maintaining the desired accuracy. However, this method is built upon a uniform grid hierarchy, which makes it difficult to exploit the spatially localized features of the solution. This limitation becomes particularly evident in the presence of singularities, as uniform refinement often results in a significant loss of efficiency. To address this issue, it is desirable to incorporate an adaptive strategy into the MLMC framework to enhance overall performance.

In this subsection, we develop an adaptive Multilevel Monte Carlo (Adaptive MLMC) algorithm, where the core idea is to embed an adaptive finite element procedure into the traditional MLMC framework so that the mesh resolution can be dynamically adjusted according to the a posteriori error indicators. Specifically, the adaptation process consists of two main stages: (i)at each level, the local behavior of the solution is detected on the mesh using the a posteriori error indicator; and (ii) the mesh is locally refined based on a chosen marking strategy (such as the maximum marking (Babuvška and Rheinboldt, 1978) or the $Dörfler$ marking (Dörfler, 1996)), thus avoiding the additional cost of global uniform refinement. In addition, a level-dependent tolerance Tol_ℓ is imposed at each level to ensure that (4.1) is satisfied, preventing excessive refinement. Through this level-by-level iteration, the meshes at different levels can automatically adapt to the singular regions of the problem, achieving a better balance between accuracy and computational cost.

We first present the pathwise adaptive finite element algorithm, following (Youett, 2018), as summarized in **Algorithm 2**.

Algorithm 2 Pathwise Adaptive Finite Element Algorithm

1: **Input**: initial mesh $\mathcal{T}_{(0)}$, sample ω, marking parameter $\theta \in (0,1)$, tolerance ε
2: Initialize $\tilde{\mathcal{T}}_{(0)}(\omega) := \mathcal{T}_{(0)}$, $k = 0$
3: **while** true **do**
4: Solve (2.2) on $\tilde{\mathcal{T}}_{(k)}(\omega)$ to obtain the finite element approximation $\tilde{\lambda}_{(k)}(\omega)$ and $\tilde{u}_{(k)}(\omega)$
5: Compute the local error indicators $\eta_T(\omega, \tilde{\lambda}_{(k)}\tilde{u}_{(k)})$ for all $T \in \tilde{\mathcal{T}}_{(k)}(\omega)$
6: **if** $\eta^2_{\tilde{\mathcal{T}}_{(k)}}(\omega, \tilde{\lambda}_{(k)}\tilde{u}_{(k)}) \leq \varepsilon$ **then**
7: Terminate the algorithm
8: **end if**
9: Determine the subset $\mathcal{M}_{(k)}(\omega) \subset \tilde{\mathcal{T}}_{(k)}(\omega)$ with the smallest cardinality, such that

$$\theta \eta^2_{\tilde{\mathcal{T}}_{(k)}}(\omega, \tilde{\lambda}_{(k)}\tilde{u}_{(k)}) \leq \sum_{T \in \mathcal{M}_{(k)}(\omega)} \eta^2_T(\omega, \tilde{\lambda}_{(k)}\tilde{u}_{(k)})$$

10: Refine all $T \in \mathcal{M}_{(k)}(\omega)$ to generate the new mesh $\tilde{\mathcal{T}}_{(k+1)}(\omega)$ and set $k := k+1$
11: **end while**
12: **Output**: $\tilde{\lambda}_{(k)}(\omega)$, $k_{\text{end}} = k$

In Steps 9-10 of **Algorithm 2**, the $Dörfler$ marking strategy (Dörfler, 1996) with parameter θ is employed, and the refinement is performed using the newest-vertex bisection technique (Binev et al., 2004; Chen and Zhang, 2010).

By combining **Algorithm 1** with **Algorithm 2**, we obtain the adaptive Multilevel Monte Carlo algorithm, as summarized in **Algorithm 3**.

Algorithm 3 Adaptive Multilevel Monte Carlo Algorithm

Input: $\mathcal{T}_{(0)}$, Tol_0, q, Tol, N_{init}, θ
Initialize $L := 1$, set $N_\ell := N_{\text{init}}$, $\ell = 0, \ldots, L$
Generate N_ℓ samples and set $Tol_\ell := q^\ell Tol_0$, $\ell = 0, \ldots, L$
Apply **Algorithm 2** with $\varepsilon = Tol_0$ to compute $\tilde{\lambda}_0(\omega_i)$ for $i = 1, \ldots, N_0$
Apply **Algorithm 2** with $\varepsilon = Tol_1$ to compute $\tilde{\lambda}_1(\omega_i)$ and with $\varepsilon = Tol_0$ to compute $\tilde{\lambda}_0(\omega_i)$ for $i = 1, \ldots, N_1$
while true **do**
    while $L > 1$ **do**
        Generate $N_L := N_{\text{init}}$ samples; $Tol_\ell := q^\ell Tol_0$, $\ell = L-1, L$
        Apply **Algorithm 2** with $\varepsilon = Tol_L$ to compute $\tilde{\lambda}_L(\omega_i)$ and with $\varepsilon = Tol_{L-1}$ to compute $\tilde{\lambda}_{L-1}(\omega_i)$ for $i = 1, \ldots, N_L$
    end while
    Compute/update $\mathbb{V}[\tilde{\lambda}_\ell(\omega) - \tilde{\lambda}_{\ell-1}(\omega)]$ and determine the optimal sample numbers N_ℓ^{opt} according to (3.8), $\ell = 0, \ldots, L$, with $\tilde{\lambda}_{-1}(\omega) = 0$
    if $N_\ell^{\text{opt}} > N_\ell$ **then**
        Generate additional $N_\ell^{\text{opt}} - N_\ell$ samples for $\ell = 0, \ldots, L$
        Apply **Algorithm 2** with $\varepsilon = Tol_\ell$ to compute $\tilde{\lambda}_\ell(\omega_i)$, and apply **Algorithm 2** with $\varepsilon = Tol_{\ell-1}$ to compute $\tilde{\lambda}_{\ell-1}(\omega_i)$ for $\ell = 1, \ldots, L$ and $i = 1, \ldots, N_\ell^{\text{opt}} - N_\ell$; for $\ell = 0$, apply **Algorithm 2** with $\varepsilon = Tol_0$ to compute $\tilde{\lambda}_0(\omega_i)$ for $i = 1, \ldots, N_0^{\text{opt}} - N_0$.
    end if
    Set $N_\ell := N_\ell^{\text{opt}}$
    if the convergence criterion (3.7) is satisfied **then**
        Terminate the algorithm
    else
        Set $L := L + 1$
    end if
end while
Compute the final estimator $\mathbb{E}[\tilde{\lambda}_L]$ according to (3.2)
Output: $\mathbb{E}[\tilde{\lambda}_L]$

Algorithm 3 employs the adaptive finite element method at each level to compute approximate eigenvalues and uses the optimized sample allocation to control the variance, thereby achieving the goal of reducing the total computational costs while meeting the preset accuracy requirements.

Following Corollary 1 of (Cui et al., 2024), the adaptive MLMC method can be shown to satisfy the convergence properties of both expectation and variance, as stated in Theorem 3.1.

Theorem 3.1. *Let* $(\tilde{\lambda}_\ell(\omega), \tilde{u}_\ell(\omega))$ *and* $(\tilde{\lambda}_{\ell-1}(\omega), \tilde{u}_{\ell-1}(\omega))$ *be the approximate eigenpairs obtained by*

***Algorithm 3.3**. Then there hold*

$$\left|\mathbb{E}\left[\tilde{\lambda}_\ell(\omega) - \tilde{\lambda}_{\ell-1}(\omega)\right]\right| \leq c_1 \cdot M_{h_\ell}^{-s},$$

$$\mathbb{V}\left[\tilde{\lambda}_\ell(\omega) - \tilde{\lambda}_{\ell-1}(\omega)\right] \leq c_2 \cdot M_{h_\ell}^{-2s},$$

where c_1 and c_2 are positive constants.

By replacing λ_ℓ in (3.3) with $\tilde{\lambda}_\ell$ obtained from Algorithm 3.3, we get the approximate solution $\tilde{Y}$ obtained by the adaptive MLMC algorithm. Using the analytical methods in (Cliffe et al., 2011; Giles, 2015), we can establish the overall computational complexity theorem for the adaptive MLMC algorithm.

Theorem 3.2. *Under **Assumption 3**, let $\tilde{Y}$ denote the approximation obtained by **Algorithm 3.3**. Then, for any preset tolerance Tol > 0, there exist a constant c, a level $L \in \mathbb{N}$, and a sequence of integers $\{N_\ell\}_{\ell=0}^{L}$ such that*

$$\mathit{MSE}(\mathbb{E}[\lambda(\omega)], \tilde{Y}) \leq \mathit{Tol}^2,$$

and the total computational cost C_{total} satisfies

$$C_{\mathit{total}} \leq \begin{cases} c \cdot \mathit{Tol}^{-2}, & \gamma < 2s, \\ c \cdot \mathit{Tol}^{-2}(\log \mathit{Tol})^2, & \gamma = 2s, \\ c \cdot \mathit{Tol}^{-2-\frac{\gamma-2s}{s}}, & \gamma > 2s. \end{cases}$$

4 Numerical Experiments

In this section we will present some numerical experiments to evaluate the proposed adaptive MLMC algorithm. The efficiency and robustness of the algorithm will be verified by comparing it with the traditional MLMC method. All experiments compare the performance differences between the two methods under the same preset accuracy Tol. In the MLMC method, the meshes are generated by uniform refinement from the initial mesh $\mathcal{T}_{(0)}$ with $h_0 = 2^{-3}$; whereas in the adaptive MLMC method, the grid starts from an initial mesh consisting of 128 congruent triangles, and mesh refinement is carried out according to the a posteriori error indicators and the tolerance recursion $Tol_\ell = \frac{1}{2}Tol_{\ell-1}$. For the adaptive MLMC algorithm (Algorithm 3.3), we set

$$Tol_0 := \|\eta^2_{\mathcal{T}_{(0)}}(\cdot, \tilde{\lambda}_{(0)}\tilde{u}_{(0)})\|_{L^2(\Omega)},$$

where the $L^2(\Omega)$ norm is computed using 1000 Monte Carlo samples. The mesh refinement process employs the Dörfler marking strategy with $\theta = 0.4$, and the initial sample size at each level is set to $N_{\text{init}} = 50$. For statistical comparison, the average degrees of freedom (DoFs) per level in the adaptive MLMC method are rounded to ensure comparability with the uniform refinement strategy used in the standard MLMC method. By analyzing the sample distributions across different levels, the variation in DoFs, and the total computational cost

$$\sum_{\ell=0}^{L}\sum_{i=1}^{N_\ell} M_{h_\ell}(\omega_i),$$

we comprehensively demonstrate the performance of the adaptive MLMC in various domains, coefficient types, and in cases where the problem exhibits singularities. For the total cost evaluation, the computational cost per sample at level ℓ is taken as $C_\ell = M_{h_\ell}$.

In our tables, we only list the computational results corresponding to each preset accuracy, without showing the tolerance sequence in detail. This is because all experiments are conducted

under the constraint of meeting the preset accuracy, and the evolution of the tolerance sequence is already implicitly encoded in the tolerance control and adaptive strategy. Reporting only the prescribed accuracies highlights the advantages of the proposed algorithm in terms of computational efficiency, DoFs, and sample size, thus providing a clearer comparison of the overall performance between the two methods.

4.1 Experiment 1

In this experiment, we solve the problem (2.1) on the unit square domain $D_s = [0,1]^2$ and the L-shaped domain $D_L = [0,1]^2 \setminus \left((\frac{1}{2},1) \times (0,\frac{1}{2})\right)$, where the coefficient is chosen as an oscillatory function:

$$a = 0.1 + (\sin x_1 \cos x_2) \cdot \omega, \quad \omega \sim \mathcal{U}(0,1), \quad b = 0.$$

The experiment compares the MLMC and the adaptive MLMC in terms of the distributions of the average DoFs and the sample numbers N_ℓ across different levels ℓ, to assess the effectiveness of the adaptive strategy in reducing the computational cost while maintaining the desired accuracy.

Table 1: Computation statistics of MLMC and Adaptive MLMC on the unit square domain when $Tol = 0.01$ ($a = 0.1 + (\sin x_1 \cos x_2) \cdot \omega$)

Method	Level ℓ	DoFs	N_ℓ	Total Samples
MLMC	0	49	77,735	79,358
	1	225	1,411	
	2	961	181	
	3	3,969	27	
	4	16,129	4	
Adaptive MLMC	0	49	73,803	75,278
	1	166	1,342	
	2	692	109	
	3	3,372	19	
	4	12,522	5	

Table 1 lists the DoFs and sample sizes for both methods at each level under the preset accuracy $Tol = 0.01$, from which it can be seen that as the level increases, the DoFs grow exponentially, while the sample size decreases gradually. Compared with MLMC, the adaptive MLMC exhibits fewer DoFs and sample sizes at all levels, thus reducing the total computational cost while maintaining numerical accuracy.

Table 2 summarizes the total computational costs and approximate expectation values for different prescribed accuracies Tol. The results indicate that the adaptive MLMC saves computational effort compared with MLMC, especially under high-accuracy conditions, demonstrating its favorable computational efficiency.

Table 3 presents the distributions of DoFs and sample numbers across levels for the two methods on the L-shaped domain. Similar to the unit square case, it can be observed from Table 3 that the DoFs grow exponentially with the level. Under the same prescribed accuracy $Tol = 0.01$, the DoFs and total sample size of the adaptive MLMC are both lower than those of MLMC, thereby effectively reducing computational costs while maintaining accuracy.

Table 2: Computational costs and approximate expectation values comparison between MLMC and Adaptive MLMC on the unit square domain under different preset accuracies ($a = 0.1 + (\sin x_1 \cos x_2) \cdot \omega$)

Tol	MLMC		Adaptive MLMC	
	C_{total}	$\mathbb{E}[Y]$	C_{total}	$\mathbb{E}[\tilde{Y}]$
0.5	$2,382$	4.6972	$1,613$	4.6596
0.1	$43,057$	5.0472	$38,946$	5.0555
0.05	$179,028$	4.9911	$166,481$	5.0539
0.01	$4,472,110$	4.9690	$4,076,046$	5.0397

Table 3: Computation statistics of MLMC and Adaptive MLMC on the L-shaped domain when $Tol = 0.01$ ($a = 0.1 + (\sin x_1 \cos x_2) \cdot \omega$)

Method	Level ℓ	DoFs	N_ℓ	Total Samples
MLMC	0	161	$119,047$	$120,385$
	1	705	$1,148$	
	2	$2,945$	159	
	3	$12,033$	26	
	4	$48,641$	5	
Adaptive MLMC	0	161	$117,661$	$119,298$
	1	506	$1,416$	
	2	$2,097$	189	
	3	$9,352$	27	
	4	$36,308$	5	

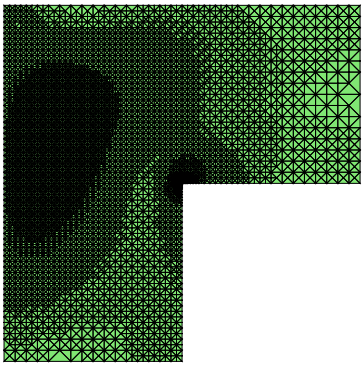


Figure 1: Example of adaptive refinement mesh on the L-shaped domain in Experiment 1.

Figure 1 shows an example of the adaptively refined mesh on the L-shaped domain. Significant mesh refinement can be observed near the reentrant corner, effectively capturing the local singular behavior of the solution.

Table 4: Comparison of total computational cost and approximate expectation values of MLMC and Adaptive MLMC on the L-shaped domain under different prescribed accuracies ($a = 0.1 + (\sin x_1 \cos x_2) \cdot \omega$)

Tol	MLMC		Adaptive MLMC	
	C_{total}	$\mathbb{E}[Y]$	C_{total}	$\mathbb{E}[\tilde{Y}]$
0.5	$10,351$	7.7959	$8,062$	7.7618
0.1	$204,538$	7.8790	$199,080$	7.9124
0.05	$837,031$	7.8333	$826,951$	7.8802
0.01	$21,000,225$	7.8298	$20,644,090$	7.8738

Table 4 summarizes the total computational costs and approximate expectation values under different prescribed accuracies on the L-shaped domain. The results further confirm that the adaptive MLMC maintains its efficiency advantage on non-convex domain. In particular, under high-accuracy tolerances, the adaptive method continues to demonstrate significant computational savings.

4.2 Experiment 2

In this experiment, we select a more complex exponential-type random function as the coefficient compared with Experiment 1:

$$a = \exp\left(\sum_i \omega_i \, k(\mathbf{x} - \mathbf{c}_i)\right), \quad b = 0,$$

where $\omega_i \sim \mathcal{U}(0,1)$, and $k(\mathbf{x}-\mathbf{c}_i) = \exp\left[-\frac{25}{2}\|\mathbf{x}-\mathbf{c}_i\|_2\right]$ denotes the exponential kernel function. Here, $\mathbf{c}_i$ are the center nodes of a uniform 5×5 grid over the square domain $D_s = [0,1]^2$. For the L-shaped domain $D_L = [0,1]^2 \setminus \left((\frac{1}{2},1) \times (0,\frac{1}{2})\right)$, the kernel parameter is reduced from 25 to 19, i.e., the number of centers decreases by six. We perform the simulations separately on the unit square and the L-shaped domain, and the results are summarized in Tables 5-8.

Table 5: Computation statistics of MLMC and Adaptive MLMC on the unit square when $Tol = 0.01$ ($a = \exp(\sum_{i=1}^{25} \omega_i k(\mathbf{x}-\mathbf{c}_i))$).

Method	Level ℓ	DoFs	N_ℓ	Total Samples
MLMC	0	49	54,013	55,322
	1	225	1,126	
	2	961	159	
	3	3,969	19	
	4	16,129	4	
	5	65,025	1	
Adaptive MLMC	0	49	53,556	55,124
	1	212	1,444	
	2	913	103	
	3	3,575	17	
	4	14,326	3	
	5	57,559	1	

Table 6: Comparison of computational cost and approximate expectation for different prescribed tolerances on the unit square ($a = \exp(\sum_{i=1}^{25} \omega_i k(\mathbf{x}-\mathbf{c}_i))$).

Tol	MLMC		Adaptive MLMC	
	C_{total}	$\mathbb{E}[Y]$	C_{total}	$\mathbb{E}[\tilde{Y}]$
0.5	2,264	30.0555	2,414	30.1194
0.1	35,522	29.9465	34,445	30.0797
0.05	146,396	29.9045	142,940	30.0327
0.01	3,257,738	29.9016	3,197,800	30.0231

Table 5 reports the average DoFs and sample sizes at different levels for the two methods. Under the preset tolerance $Tol = 0.01$, the adaptive MLMC shows a reduction in both DoFs and total sample counts compared to the MLMC, although the improvement is less pronounced than that in **Experiment 1**. As can be seen from Table 6, the adaptive MLMC incurs slightly higher computational cost at low accuracy ($Tol = 0.5$), but consistently outperforms the MLMC as the tolerance decreases to $Tol = 0.1, 0.05$, and 0.01, demonstrating its advantage for high-accuracy computations.

For the L-shaped domain, we additionally provide a visualization of the adaptive mesh refinement as shown in Fig. 2. Due to the geometric singularity and random coefficient structure, the mesh is strongly refined near the reentrant corners and high-gradient regions, enabling more accurate

resolution of localized features. This illustrates the capability of the adaptive MLMC to reduce error effectively through adaptive refinement on complex domains.

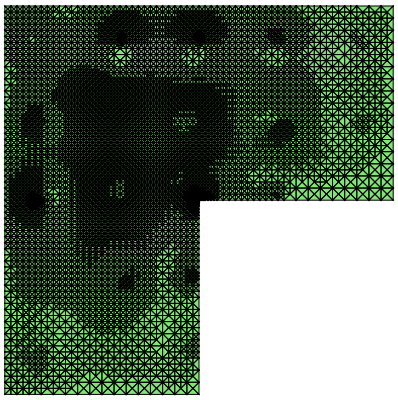


Figure 2: Example of adaptive mesh refinement on the L-shaped domain for Experiment 2.

Tables 7 and 8 summarize the results on the L-shaped domain. Table 7 also reports the average DoFs and sample distribution at different levels when $Tol = 0.01$, from which one can see that the adaptive MLMC exhibits fewer DoFs, a smaller total sample count, and fewer levels than the MLMC, thereby reducing the total computational cost while maintaining accuracy. Combined with the cost comparison in Table 8, it can be observed that the adaptive MLMC consistently outperforms the MLMC as the prescribed tolerance tightens, highlighting its efficiency on complex domains and under high-accuracy requirements.

4.3 Experiment 3

To further investigate the performance of the adaptive MLMC method for problems with singularities, we introduce a locally singular source term to the problem (1.1)-(1.2), resulting in the following formulation:

$$-\nabla \cdot (a(\mathbf{x},\omega)\nabla u(\mathbf{x},\omega)) + b(\mathbf{x},\omega)u(\mathbf{x},\omega) = \lambda(\omega)c(\mathbf{x},\omega)u(\mathbf{x},\omega), \quad \text{in } \mathbf{x} \in D, \tag{4.1}$$

$$u(\mathbf{x},\omega) = 0, \qquad \text{on } \mathbf{x} \in \Gamma := \partial D, \tag{4.2}$$

where $D = [-1,1]^2$ and

$$a = 1, \quad b = 0, \quad c = \exp\big(-100|\mathbf{x} - Y(\omega)|^2\big),$$

with $Y(\omega) = (\omega_1, \omega_2)^T$ be a random vector with components independently drawn from $\mathcal{U}(-0.25, 0.25)$. This configuration introduces a localized singular perturbation at random points, thereby providing a stringent test of the algorithm under nonsmooth conditions.

To illustrate the influence of the random source term within the singular region, Fig. 3 presents a visualization of the approximate eigenfunction associated with the smallest eigenvalue for one sample. The figure clearly shows a high-amplitude peak generated by the source term at a random location, which induces strong local variations and pronounced nonsmoothness in the solution. Such features pose significant challenges for numerical algorithms.

Table 7: Computation statistics of MLMC and Adaptive MLMC on the L-shaped domain when $Tol = 0.01$ ($a = \exp(\sum_{i=1}^{19} \omega_i k(\mathbf{x} - \mathbf{c}_i))$).

Method	Level ℓ	DoFs	M_ℓ	Total Samples
MLMC	0	161	267,236	271,165
	1	705	3,396	
	2	2,945	438	
	3	12,033	78	
	4	48,641	13	
	5	195,585	3	
	6	784,385	1	
Adaptive MLMC	0	161	266,311	270,782
	1	611	3,587	
	2	2,532	774	
	3	10,499	94	
	4	40,431	14	
	5	166,377	2	

Table 8: Comparison of computational cost and approximate expectation for different prescribed tolerances on the L-shaped domain ($a = \exp(\sum_{i=1}^{19} \omega_i k(\mathbf{x} - \mathbf{c}_i))$).

Tol	MLMC		Adaptive MLMC	
	C_{total}	$\mathbb{E}[Y]$	C_{total}	$\mathbb{E}[\tilde{Y}]$
0.5	20,938	58.1930	20,498	58.3297
0.1	534,671	57.3071	533,960	57.4733
0.05	2,150,102	57.3728	2,010,954	57.5329
0.01	49,651,133	57.3507	48,957,639	57.5033

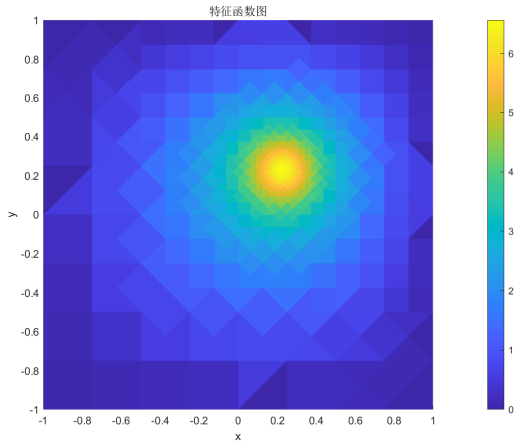
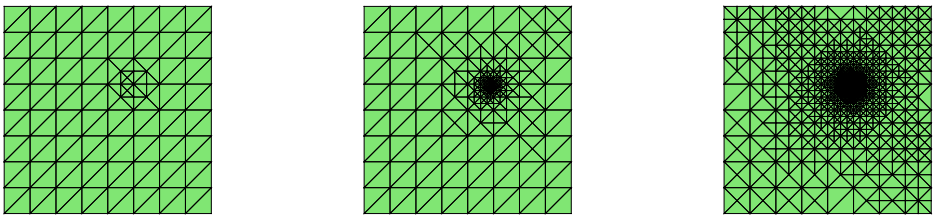


Figure 3: Example of the eigenfunction corresponding to the approximation of the smallest eigenvalue in Experiment 3.

To further demonstrate the adaptivity of the method in the vicinity of the singularity, Fig. 4 depicts the evolution of the adaptive mesh during successive refinement steps.



(a) Five refinements. (b) Ten refinements. (c) Fifteen refinements.

Figure 4: Adaptive mesh refinement at different stages. Significant refinement occurs near the peak region, while coarser meshes are maintained elsewhere.

As shown in Fig. 4, the mesh is strongly refined in regions surrounding the source peak and remains coarse away from it. This localized refinement substantially improves the numerical accuracy near the singular region while avoiding excessive computational costs across the entire domain.

In addition to these visualizations, Table 9 compares the distributions of average DoFs and sample sizes between the MLMC and the adaptive MLMC. Similar to the results in **Experiments 1 and 2**, Table 9 shows that as the level ℓ increases, the DoFs grow exponentially, while the sample size decreases gradually. Notably, under the tolerance $Tol = 0.01$, the adaptive MLMC method uses more levels than the MLMC, but achieves significantly lower average DoFs and total sample counts. This improvement is more pronounced than in **Experiments 1 and 2**, underscoring the superiority of the adaptive MLMC for singular problems.

Table 10 presents the total computational costs and approximate expectation values for the adaptive MLMC and the MLMC methods under different tolerances. As the tolerance Tol decreases,

Table 9: Computation statistics of MLMC and Adaptive MLMC for the singular problem (4.1)-(4.2) when $Tol = 0.01$

Method	Level ℓ	DoFs	N_ℓ	Total Samples
MLMC	0	49	7,783,218	12,957,888
	1	225	4,319,044	
	2	961	64,572	
	3	3,969	2,113	
	4	16,129	294	
	5	65,025	33	
	6	261,121	5	
	7	1,046,529	2	
Adaptive MLMC	0	49	4,106,760	6,686,760
	1	57	1,922,144	
	2	103	445,341	
	3	329	60,131	
	4	1,263	6,684	
	5	5,359	1,432	
	6	20,139	183	
	7	80,258	17	
	8	321,725	3	

Table 10: Comparison of computational cost and approximate expectation values of MLMC and Adaptive MLMC for the singular problem (4.1)-(4.2) at different prescribed tolerances Tol.

Tol	MLMC		Adaptive MLMC	
	C_{total}	$\mathbb{E}[Y]$	C_{total}	$\mathbb{E}[\tilde{Y}]$
0.5	590,108	87.1513	165,071	92.6640
0.1	14,624,911	86.4257	3,980,289	92.4610
0.05	57,524,105	86.4027	16,039,134	92.4649
0.01	1,433,889,185	86.3968	399,267,282	92.4752

the overall computational cost grows rapidly. Under all tested tolerances, the adaptive MLMC significantly outperforms the MLMC: when $Tol = 0.5$, the cost is reduced by about 72%; when $Tol = 0.01$, the reduction remains above 70%. Compared with **Experiments 1 and 2**, the advantage is more pronounced, indicating that the adaptive multilevel strategy is more effective at controlling computational effort in singular problems, demonstrating superior adaptability and robustness.

5 Conclusion

For the elliptic eigenvalue problem with random coefficients, this paper proposes and investigates an adaptive Multilevel Monte Carlo (Adaptive MLMC) method. Through systematic tests on the unit square, on an L-shaped domain with geometric singularities, and on complex benchmark problems containing locally strong singular source terms, the numerical results demonstrate that, compared with the standard MLMC method, the adaptive MLMC method effectively reduces both the average degrees of freedom and the total number of samples under different accuracy requirements. Consequently, it significantly lowers the overall computational cost while maintaining the numerical accuracy of the approximations. Under high-accuracy settings and in the presence of singularities, the advantages of the adaptive strategy become particularly evident, highlighting its stability and robustness when computing nonsmooth solutions.

References

Ainsworth, M. and Oden, J. T. (1997). A posteriori error estimation in finite element analysis. *Computer methods in applied mechanics and engineering*, 142(1-2):1–88.

Babuška, I. (1970). The finite element method for elliptic equations with discontinuous coefficients. *Computing*, 5(3):207–213.

Babuška, I. and Osborn, J. (1991). Eigenvalue problems.

Babuvška, I. and Rheinboldt, W. C. (1978). Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis*, 15(4):736–754.

Barth, A., Schwab, C., and Zollinger, N. (2011). Multi-level monte carlo finite element method for elliptic pdes with stochastic coefficients. *Numerische Mathematik*, 119(1):123–161.

Binev, P., Dahmen, W., and DeVore, R. (2004). Adaptive finite element methods with convergence rates. *Numerische Mathematik*, 97(2):219–268.

Bourlard, M., Dauge, M., Lubuma, M.-S., and Nicaise, S. (1992). Coefficients of the singularities for elliptic boundary value problems on domains with conical points. iii: Finite element methods on polygonal domains. *SIAM journal on numerical analysis*, 29(1):136–155.

Charrier, J., Scheichl, R., and Teckentrup, A. L. (2013). Finite element error analysis of elliptic pdes with random coefficients and its application to multilevel monte carlo methods. *SIAM Journal on Numerical Analysis*, 51(1):322–352.

Chen, L. and Zhang, C. (2010). A coarsening algorithm on adaptive grids by newest vertex bisection and its applications. *Journal of Computational Mathematics*, pages 767–789.

Clément, P. (1975). Approximation by finite element functions using local regularization. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):77–84.

Cliffe, K. A., Giles, M. B., Scheichl, R., and Teckentrup, A. L. (2011). Multilevel monte carlo methods and applications to elliptic pdes with random coefficients. *Computing and Visualization in Science*, 14(1):3.

Collier, N., Haji-Ali, A.-L., Nobile, F., Von Schwerin, E., and Tempone, R. (2015). A continuation multilevel monte carlo algorithm. *BIT Numerical Mathematics*, 55(2):399–432.

Cui, T., De Sterck, H., Gilbert, A. D., Polishchuk, S., and Scheichl, R. (2024). Multilevel monte carlo methods for stochastic convection–diffusion eigenvalue problems. *Journal of Scientific Computing*, 99(3):77.

Detommaso, G., Dodwell, T., and Scheichl, R. (2019). Continuous level monte carlo and sample-adaptive model hierarchies. *SIAM/ASA Journal on Uncertainty Quantification*, 7(1):93–116.

Dobson, D. C. (1999). An efficient method for band structure calculations in 2d photonic crystals. *Journal of Computational Physics*, 149(2):363–376.

Dörfler, W. (1996). A convergent adaptive algorithm for poisson's equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124.

Duderstadt, J. J. and Hamilton, L. J. (1976). *Nuclear reactor analysis*. Wiley.

Durán, R. G., Padra, C., and Rodríguez, R. (2003). A posteriori error estimates for the finite element approximation of eigenvalue problems. *Mathematical Models and Methods in Applied Sciences*, 13(08):1219–1229.

Eigel, M., Merdon, C., and Neumann, J. (2016). An adaptive multilevel monte carlo method with stochastic bounds for quantities of interest with uncertain data. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1219–1245.

Elfverson, D., Hellman, F., and Målqvist, A. (2016). A multilevel monte carlo method for computing failure probabilities. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):312–330.

Gerstner, T. and Heinz, S. (2012). Dimension-and time-adaptive multilevel monte carlo methods. In *Sparse Grids and Applications*, pages 107–120. Springer.

Giani, S. and Graham, I. G. (2012). Adaptive finite element methods for computing band gaps in photonic crystals. *Numerische Mathematik*, 121(1):31–64.

Gilbert, A. D., Graham, I. G., Kuo, F. Y., Scheichl, R., and Sloan, I. H. (2019). Analysis of quasi-monte carlo methods for elliptic eigenvalue problems with stochastic coefficients. *Numerische Mathematik*, 142(4):863–915.

Gilbert, A. D. and Scheichl, R. (2024). Multilevel quasi-monte carlo for random elliptic eigenvalue problems ii: efficient algorithms and numerical results. *IMA Journal of Numerical Analysis*, 44(1):504–535.

Giles, M. B. (2008). Multilevel monte carlo path simulation. *Operations research*, 56(3):607–617.

Giles, M. B. (2015). Multilevel monte carlo methods. *Acta numerica*, 24:259–328.

Haji-Ali, A.-L., Nobile, F., von Schwerin, E., and Tempone, R. (2016). Optimization of mesh hierarchies in multilevel monte carlo samplers. *Stochastics and Partial Differential Equations Analysis and Computations*, 4(1):76–112.

Heinrich, S. (2001). Multilevel monte carlo methods. In *International Conference on Large-Scale Scientific Computing*, pages 58–67. Springer.

Hoel, H., Von Schwerin, E., Szepessy, A., and Tempone, R. (2011). Adaptive multilevel monte carlo simulation. In *Numerical Analysis of Multiscale Computations: Proceedings of a Winter Workshop at the Banff International Research Station 2009*, pages 217–234. Springer.

Hoel, H., Von Schwerin, E., Szepessy, A., and Tempone, R. (2014). Implementation and analysis of an adaptive multilevel monte carlo algorithm. *Monte Carlo Methods and Applications*, 20(1):1–41.

Jamelot, E. and Ciarlet Jr, P. (2013). Fast non-overlapping schwarz domain decomposition methods for solving the neutron diffusion equation. *Journal of Computational Physics*, 241:445–463.

Khodadadian, A., Parvizi, M., and Heitzinger, C. (2020). An adaptive multilevel monte carlo algorithm for the stochastic drift–diffusion–poisson system. *Computer Methods in Applied Mechanics and Engineering*, 368:113163.

Kornhuber, R. and Youett, E. (2018). Adaptive multilevel monte carlo methods for stochastic variational inequalities. *SIAM Journal on Numerical Analysis*, 56(4):1987–2007.

Kuchment, P. (2001). The mathematics of photonic crystals. In *Mathematical modeling in optical science*, pages 207–272. SIAM.

Kuo, F. Y., Schwab, C., and Sloan, I. H. (2012). Quasi-monte carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM Journal on Numerical Analysis*, 50(6):3351–3374.

Norton, R. A. and Scheichl, R. (2013). Planewave expansion methods for photonic crystal fibres. *Applied numerical mathematics*, 63:88–104.

Scheichl, R. (1997). *Parallel solution of the transient multigroup neutron diffusion equations with multi-grid and preconditioned Krylov-subspace methods*. Trauner.

Teckentrup, A. L., Scheichl, R., Giles, M. B., and Ullmann, E. (2013). Further analysis of multilevel monte carlo methods for elliptic pdes with random coefficients. *Numerische Mathematik*, 125(3):569–600.

Verfürth, R. (1996). A review of a posteriori error estimation and adaptive mesh-refinement techniques. *(No Title)*.

Youett, E. (2018). *Adaptive Multilevel Monte Carlo Methods for Random Elliptic Problems*. Freie Universitaet Berlin (Germany).