AN EFFICIENT ITERATIVE METHOD FOR SOLVING PARAMETER-DEPENDENT AND RANDOM CONVECTION-DIFFUSION PROBLEMS

XIAOBING FENG*, YAN LUO†, LIET VO‡, AND ZHU WANG§

Abstract. This paper develops and analyzes a general iterative framework for solving parameter-dependent and random convection-diffusion problems. It is inspired by the multi-modes method of [7, 8] and the ensemble method of [20] and extends those methods into a more general and unified framework. The main idea of the framework is to reformulate the underlying problem into another problem with parameter-independent convection and diffusion coefficients and a parameter-dependent (and solution-dependent) right-hand side, a fixed-point iteration is then employed to compute the solution of the reformulated problem. The main benefit of the proposed approach is that an efficient direct solver and a block Krylov subspace iterative solver can be used at each iteration, allowing to reuse the LU matrix factorization or to do an efficient matrix-matrix multiplication for all parameters, which in turn results in significant computation saving. Convergence and rates of convergence are established for the iterative method both at the variational continuous level and at the finite element discrete level under some structure conditions. Several strategies for establishing reformulations of parameter-dependent and random diffusion and convection-diffusion problems are proposed and their computational complexity is analyzed. Several 1-D and 2-D numerical experiments are also provided to demonstrate the efficiency of the proposed iterative method and to validate the theoretical convergence results.

Key words. Parameter-dependent and random convection-diffusion problems, multi-modes method, ensemble method, finite element , iterative algorithm, computational complexity.

AMS subject classifications. 65N12, 65N15, 65N30

1. Introduction. Parameter-dependent problems arise from many engineering and scientific applications such as fluid mechanics, porous media flow, wave propagation and various biological models (cf. [19] and the references therein). The parameter could be deterministic (such as viscosity, diffusivity, permeability, frequency, and birth/death rate) or random, they appear as parameters in their respective mathematical (i.e., PDE) models. As the solution of such a model depends on the parameter, usually in a nonlinear manner even the PDE is linear, to compute approximate solutions, one is forced to solve the same problem multiple times for a given set of parameters. Such a direct (or brute force) approach may still be feasible if the parameter set is small even it is not efficient, however, it becomes too expensive to use if the parameter set is large due to the sheer amount of computation required to solve a complicated PDE problem tens, hundreds even thousands times.

To overcome the computational challenge, several approaches have been proposed in literature including variants of the model order reduction (MOR) method [14, 17,

---

*Department of Mathematics, The University of Tennessee, Knoxville, TN 37996, U.S.A. (xfeng@math.utk.edu) The work of this author was partially supported by the NSF grants DMS-1620168 and DMS-2012414.

†School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China (luoyan_16@126.com) The work of this author was supported by the Young Scientists Fund of the National Natural Science Foundation of China grant 11901078 and by the Fundamental Research Funds for the Central Universities grant ZGX2020J021.

‡Department of Mathematics, The University of Tennessee, Knoxville, TN 37996, U.S.A. (lvo6@vols.utk.edu The work of this author was partially supported by the NSF grants DMS-1620168 and DMS-2012414.

§Department of Mathematics, University of South Carolina, Columbia, SC 29208, U.S.A. (wangzhu@math.sc.edu) The work of this author was partially supported by the NSF grant DMS-1913073.

1, 4], the ensemble method [15, 12, 11, 13, 20, 21] and the multi-modes method [7, 8, 9]. The MOR methods aim to lower the computational complexity of the underlying model/problem by approximating the solution manifold using a handful of degrees of freedom, a low-dimensional surrogate to the original model is then built which is usually cheaper to simulate and often referred to as a reduced order model. Therefore, the computational saving is due to the dimensionality reduction. On the other hand, noticing the similarity of solving the same type problems with different parameters, a natural idea is to solve those problems together as a group and to reuse the computation as much as possible, so the computational saving is due to reusing a major portion of the computation for a set of parameters and requires designing better algorithms. This is exactly the idea used by the ensemble and multi-modes methods, and is also the approach adopted in this paper.

Inspired by the ensemble and multi-modes methods, the primary objective of this paper is to develop and analyze a more general and unified iterative framework for solving parameter-dependent (including the case of random parameters) convection-diffusion problems in an abstract variational setting. Our main idea is to reformulate a given parameter-dependent problem into another problem with parameter-independent convection and diffusion coefficients and a parameter-dependent (and solution-dependent) right-hand side, its solution is then computed using a fixed-point iteration algorithm. Since the convection and diffusion coefficients are parameter-independent and iterate-independent, while the right-hand side term is parameter-dependent and iterate-dependent, this presents an ideal set-up for us to employ an efficient direct solver and a block Krylov subspace iterative solver at each iteration for solving a linear system with multiple right-hand vectors. It is the reuse of the LU matrix factorization and efficiently to compute matrix-matrix multiplication for all parameters that results in significant computation saving, hence, leads to an efficient overall numerical method.

The remainder of this paper is organized as follows. In Section 2 we present the abstract variational setting for the parameter-dependent convection-diffusion problems to be considered in this paper. Structure conditions are stated to ensure the well-posedness of the variational problem. Such an abstract setting allows the wider applicability of the method and results of this paper, it also permits a clean and concise presentation of the convergence analysis. In Section 3 we first present our main algorithm in the variational continuous level and then establish the rate of convergence of the algorithm under some structure conditions. Section 4 presents finite element approximations of both the variational problem and the iterative algorithm. The highlight of this section is to derive the rate of convergence for the discretized algorithm. It worths noting that the finite element approximations are chosen for the preciseness and ease of presentation, the convergence results are also valid for other Galerkin-type approximations such as discontinuous Galerkin and spectral approximations. Section 5 is devoted to computational complexity analysis of the proposed algorithm and to discussing an implementation strategy of the algorithm based on a CVT (centroidal Voronoi tessellations) clustering method. Finally, in Section 6 we present three sets of numerical tests, the first one solves a parameter-dependent diffusion problem and the second solves a random diffusion problem, and the third computes a random convection-dominated convection-diffusion problem which is a randomized double-glazing problem [18]. Both 1-D and 2-D numerical test results are presented to demonstrate the efficiency of the proposed iterative method and to validate the theoretical convergence results. In particular, the last two tests show that

the proposed iterative method can successfully solve some strongly random diffusion and convection-diffusion problems which the multi-modes method fails to do.

2. Variational problem setting. Let H be a Hilbert space and $V \subset H$ be a Banach space. V^* denotes the dual space of V, the space of bounded linear functionals on V. We consider the following variational problem of seeking $u \in V$ such that

$$(2.1) \qquad A(\omega; u, v) = F(\omega; v) \qquad \forall v \in V,$$

where $F(\omega; \cdot) \in V^*$ and $A(\omega; \cdot, \cdot) : V \times V \to \mathbb{R}$ depends on a parameter ω, which belongs to a (given) parameter space Ω and satisfies the following properties:

  (i) *Bi-linearity:* A is bilinear mapping on $V \times V$.

  (ii) *Boundedness:* There exists an ω-independent constant $\Lambda > 0$ such that

$$|A(\omega; v, w)| \leq \Lambda \|v\|_V \|w\|_V \qquad \forall v, w \in V.$$

  (iii) *Coercivity:* There is an ω-independent constant $\lambda > 0$ such that

$$A(\omega; v, v) \geq \lambda \|v\|_V^2 \qquad \forall v \in V.$$

REMARK 1.

(a) *Since $A(\omega, \cdot, \cdot)$ may be non-symmetric, then problem (2.1) contains both coercive diffusion and coercive convection-diffusion problems which could be convection-dominated, see Section 6.3-6.4 for more discussions.*

(b) *The parameter space Ω can be a deterministic or probability space, in the latter case, equation (2.1) becomes a random PDE. Moreover, Ω can be finite or infinite set.*

(c) *Applicable examples to be considered later are (i) parameter-dependent PDEs; (ii) random PDE.*

It is well known [3] that under the assumptions (i)–(iii), there holds the following Lax-Milgram Theorem.

PROPOSITION 2.1. *Under the above assumptions, problem (2.1) has a unique solution $u(\omega) \in V$ for each $\omega \in \Omega$ and there holds*

$$(2.2) \qquad \|u(\omega)\|_V \leq \frac{\|F(\omega)\|_{V^*}}{\lambda} \qquad \forall \omega \in \Omega.$$

REMARK 2. *We note that $\|F(\omega)\|_{V^*} < \infty$ for each $\omega \in \Omega$, however, the bound may depend on ω. In the subsequent sections, unless stated otherwise, we shall assume that there exists an ω-independent constant $C_F > 0$ such that $\|F(\omega)\|_{V^*} \leq C_F$ for all $\omega \in \Omega$, that is, the operator norm of $F(\omega)$ is uniformly bounded in Ω. This condition will be relaxed to $E[\|F(\cdot)\|_{V^*}] < \infty$ for random diffusion and convection-diffusion applications in Section 6, namely, the expected value of the operator norm of $F(\omega)$ is bounded.*

As mentioned earlier, the aim of this paper is to develop efficient numerical methods/algorithms for computing the solution set $u(\omega)$ for all $\omega \in \Omega$.

3. The proposed iterative framework. The goal of this section is to formulate our main iterative algorithm for solving problem (2.1) and to establish the rate of convergence for the algorithm.

3.1. Formulation of the main algorithm. Let $A_0(\cdot, \cdot) : V \times V \to \mathbb{R}$ be an ω-independent bilinear form, and set $A_1 = A - A_0$. Trivially

$$A(\omega; \cdot, \cdot) = A_0(\cdot, \cdot) + A_1(\omega; \cdot, \cdot) \tag{3.1}$$

is also a bilinear form on $V \times V$. The equation (2.1) can be rewritten as

$$A_0(u, v) = F(\omega; v) - A_1(\omega; u, v). \tag{3.2}$$

We now are ready to state our main iterative algorithm for solving (2.1).

Algorithm 1: Main Algorithm in Variational Setting

Step 1: Find $U_0 \in V$ by solving the following problem:

$$A_0(U_0, v) = F(\omega; v) \qquad \forall v \in V. \tag{3.3}$$

Step 2: For each $\omega \in \Omega$, determine $\{U_n = U_n(\omega)\}_{n \geq 1} \subset V$ recursively by solving

$$A_0(U_n, v) = F(\omega; v) - A_1(\omega; U_{n-1}, v) \qquad \forall v \in V. \tag{3.4}$$

To complete the construction of Algorithm 1, we need to specify the bilinear form A_0. As expected, it must be problem-dependent, hence, we will do so in Section 6 when we apply Algorithm 1 to specific application problems. We conclude this subsection with the following remarks.

REMARK 3. *The main advantage of Algorithm 1 is that the left hand-side bilinear forms in (3.3) and (3.4) are the same and independent of the parameter ω which allows to design fast solvers for computing the solutions in both steps of Algorithm 1.*

REMARK 4. *(a) To recover the multi-modes method of [7] for random diffusion problems with the diffusion coefficients of the form $a(\omega, x) = a_0(x) + \eta(\omega, x)$ (see Section 6.2 for the details), let $a_0(x)$ be the expected value of $a(\omega, x)$ and we further assume that $\eta = \varepsilon \xi$, then define the nth mode function by*

$$u_0 = U_0, \qquad u_n := \frac{1}{\varepsilon^n}(U_n - U_{n-1}) \quad \text{for } n \geq 1, \tag{3.5}$$

which implies that

$$U_n = U_{n-1} + \varepsilon^n u_n = u_0 + \varepsilon u_1 + \varepsilon^2 u_2 \cdots + \varepsilon^n u_n.$$

Thus, the multi-modes method of [7] is recovered. We note that the multi-modes method of [7] does compute the mode functions $\{u_j\}_{j=1}^n$ in order to form the final approximate U_n. On the other hand, Algorithm 1 does not compute the mode functions, instead, it computes the approximate U_n directly. We also note that the parameter space Ω does not need to be given a priori but can be sampled on fly in simulations.

(b) To recover the ensemble method of [20] for parameter-dependent problems with the leading-term coefficients $a(\omega_j, x)$ and the parameter set $\Omega = \{\omega_j\}_{j=1}^J$ which are a priori given (see Section 6.1 for the details), we simply choose a_0 to be

$$a_0(x) = \frac{1}{J}\sum_{j=1}^J a(\omega_j, x) \quad or \quad a_0(x) = \max_{1 \leq j \leq J} a(\omega_j, x) \qquad \forall x \in D,$$

Algorithm 1 then leads to the ensemble method.

3.2. Convergence analysis. To ensure the convergence of Algorithm 1, we impose the following criterion (and practical guideline) for choosing A_0 which would ensure the convergence of the algorithm.

Criterion for selecting A_0

(a) A_0 must satisfy (i), (ii), (iii) from definition of A. That is,
    (i) A_0 is bilinear on $V \times V$.
    (ii) There exists a constant $\Lambda_0 > 0$ such that

$$|A_0(v, w)| \leq \Lambda_0 \|v\|_V \|w\|_V \qquad \forall v, w \in V.$$

    (iii) There is a constant $\lambda_0 > 0$ such that

$$A_0(v, v) \geq \lambda_0 \|v\|_V^2 \qquad \forall v \in V.$$

(b) *Relative dominance*: There exists a $\rho \in (0, 1)$ such that

$$(3.6) \qquad \frac{\|A_1(\omega)\|_{\mathcal{L}(V \times V, \mathbb{R})}}{\lambda_0} < \rho \qquad \forall \omega \in \Omega.$$

Under the above assumptions, we have the following convergence theorem.

THEOREM 3.1. *Let $u(\omega)$ be the solution of (2.1) and $\{U_n(\omega)\}_{n \geq 0}$ be generated by Algorithm 1. Suppose the conditions of the above criterion hold. Then there exists a constant $C > 0$ independent of Ω such that*

$$(3.7) \qquad \|u(\omega) - U_n(\omega)\|_V \leq C\rho^{n+1} \qquad \forall \omega \in \Omega.$$

Hence, $U_n(\omega)$ converges to $u(\omega)$ strongly in V as $n \to \infty$ for every $\omega \in \Omega$.

Proof. For each fixed $\omega \in \Omega$, let $r_n = r_n(\omega) = u(\omega) - U_n(\omega) \in V$ for $n \geq 0$. Then, subtracting equation (3.3) from (3.2) yields the following error equation for $n = 0$:

$$(3.8) \qquad A_0(r_0, v) = -A_1(\omega; u, v) \qquad \forall v \in V.$$

Similarly, for any $n \geq 1$ subtracting (3.4) from (3.2) gives the error equation

$$(3.9) \qquad A_0(r_n, v) = -A_1(\omega; r_{n-1}, v) \qquad \forall v \in V.$$

Choose $v = r_0$ in (3.8) we obtain:

$$(3.10) \qquad \lambda_0 \|r_0\|_V^2 \leq A_0(r_0, r_0) = -A_1(\omega; u, r_0) \leq \|A_1(\omega)\|_{\mathcal{L}(V \times V, \mathbb{R})} \|u\|_V \|r_0\|_V.$$

This implies that

$$(3.11) \qquad \|r_0\|_V \leq \frac{\|A_1(\omega)\|_{\mathcal{L}(V \times V, \mathbb{R})}}{\lambda_0} \|u\|_V \leq \rho \|u\|_V.$$

We also choose $v = r_n$ in (3.9) to obtain:

$$(3.12) \qquad \lambda_0 \|r_n\|_V^2 \leq A_0(r_n, r_n) = -A_1(\omega; r_{n-1}, r_n)$$
$$\leq \|A_1(\omega)\|_{\mathcal{L}(V \times V, \mathbb{R})} \|r_{n-1}\|_V \|r_n\|_V.$$

Therefore,

$$(3.13) \qquad \|r_n\|_V \leq \frac{\|A_1(\omega)\|_{\mathcal{L}(V \times V, \mathbb{R})}}{\lambda_0} \|r_{n-1}\|_V \leq \rho \|r_{n-1}\|_V.$$

By induction we have

$$(3.14) \qquad \|r_n\|_V \leq \rho^{n+1} \|u\|_V \leq \rho^{n+1} \frac{1}{\lambda} \|F\|_{V^*}.$$

The proof is complete. □

4. Finite element approximation. In this section, we first formulate the finite element Galerkin approximation of (2.1) and then present a discrete analogue of Algorithm 1 as a fast solver for computing the finite element solutions for all $\omega \in \Omega$. To the end, let $\mathcal{T}_h$ be a quasi-uniform triangulation of a bounded domain $D \subset \mathbb{R}^d$ ($d = 1, 2, 3$) with mesh size $h \in (0, 1)$. Let V_r^h ($r \geq 1$) denote the finite element space consisting of continuous piecewise rth order polynomials associated with $\mathcal{T}_h$. We assume that V_r^h is a subspace of V, which implicitly assumes that $V \subset W^{1,1}(D)$ but $V \not\subset W^{2,1}(D)$.

For each $\omega \in \Omega$, the standard finite element Galerkin approximation of (2.1) is defined as seeking $u_h = u_h(\omega) \in V_r^h$ such that

$$(4.1) \qquad A\big(\omega; u_h, v_h\big) = F(\omega; v_h) \qquad \forall v_h \in V_r^h.$$

It is easy to show that (4.1) has a unique solution u_h which also satisfies the stability estimate (2.2). Moreover, there holds the following error estimate.

PROPOSITION 4.1. *Let $u(\omega)$ and $u_h(\omega)$ denote the solutions of (2.1) and (4.1) respectively. Then, there exists a constant $C = C(\omega) > 0$ independent of h and a positive integer $\ell(\leq r)$ such that*

$$(4.2) \qquad \|u(\omega) - u_h(\omega)\|_V \leq C h^\ell \qquad \forall \omega \in \Omega.$$

REMARK 5. *The dependence of C on ω is through the norm $\|u(\omega)\|_{H^{r+1}}$ and ℓ depends on the space V. For example, if $V = H_0^1(D)$, then $\ell = r$.*

Using the definition of A_0 and A_1, we can rewrite (4.1) as follows:

$$(4.3) \qquad A_0\big(u_h, v_h\big) = F(\omega; v_h) - A_1\big(\omega; u_h, v_h\big) \qquad \forall v_h \in V_r^h.$$

We then can easily formulate the discrete counterpart of Algorithm 1 for computing the solution of (4.1) below.

Algorithm 2: Main Algorithm in Discrete Setting

Step 1: Find $U_0^h \in V_r^h$ by solving the following problem:

$$(4.4) \qquad\qquad A_0\big(U_0^h, v_h\big) = F(\omega; v_h) \qquad \forall v_h \in V_r^h.$$

Step 2: For each $\omega \in \Omega$, determine $\{U_n^h = U_n^h(\omega)\}_{n \geq 1} \subset V_r^h$ recursively by solving

$$(4.5) \qquad A_0\big(U_n^h, v_h\big) = F(\omega; v_h) - A_1\big(\omega; U_{n-1}^h, v_h\big) \qquad \forall v_h \in V_r^h.$$

REMARK 6. *We note that the main advantage of Algorithm 2 is that the left hand-side bilinear forms (or stiffness matrices) in (4.4) and (4.5) are the same and independent of the parameter ω which allows to design fast solvers for computing the solutions in both steps of Algorithm 2.*

Similarly, there also holds the following convergence result for Algorithm 2.

THEOREM 4.2. *Let $u_h(\omega)$ be solution of (4.3) and $\{U_n^h(\omega)\}_{n \geq 0}$ be generated by Algorithm 2. Then there exists a constant $C > 0$ independent of ω such that*

$$(4.6) \qquad \|u_h(\omega) - U_n^h(\omega)\|_V \leq C\rho^{n+1} \qquad \forall \omega \in \Omega.$$

Hence, $U_n^h(\omega)$ converges to $u^h(\omega)$ strongly in V as $n \to \infty$ for every $\omega \in \Omega$.

Proof. For each fixed $\omega \in \Omega$, let $e_n^h = u_h(\omega) - U_n^h(\omega) \in V_r^h$ for $n \geq 0$. Then, subtracting equation (4.4) from (4.3) yields the following error equation for $n = 0$:

$$(4.7) \qquad A_0(e_0^h, v_h) = -A_1(\omega; u_h, v_h) \qquad \forall v_h \in V_r^h.$$

Similarly, by subtracting (4.5) from (4.3), we obtain the following error equation for $n \geq 1$:

$$(4.8) \qquad A_0(e_n^h, v_h) = -A_1(\omega; e_{n-1}^h, v_h) \qquad \forall v_h \in V_r^h.$$

Choosing $v_h = e_0^h \in V_r^h$ in (4.7), we have

$$(4.9) \qquad \lambda_0 \|e_0^h\|_V^2 \leq A_0(e_0^h, e_0^h) = -A_1(\omega; u_h, e_0^h) \leq \|A_1(\omega)\|_{\mathcal{L}(V \times V, \mathbb{R})} \|u_h\|_V \|e_0^h\|_V.$$

This implies that

$$(4.10) \qquad \|e_0^h\|_V \leq \frac{\|A_1(\omega)\|_{\mathcal{L}(V \times V, \mathbb{R})}}{\lambda_0} \|u_h\|_V \leq \rho \|u_h\|_V.$$

In addition, setting $v_h = e_n^h \in V_r^h$ in (4.8) yields

$$(4.11) \qquad \lambda_0 \|e_n^h\|_V^2 \leq A_0(e_n^h, e_n^h) = -A_1(\omega; e_{n-1}^h, e_n^h)$$
$$\leq \|A_1(\omega)\|_{\mathcal{L}(V \times V, \mathbb{R})} \|e_{n-1}^h\|_V \|e_n^h\|_V.$$

Therefore,

$$(4.12) \qquad \|e_n^h\|_V \leq \frac{\|A_1(\omega)\|_{\mathcal{L}(V \times V, \mathbb{R})}}{\lambda_0} \|e_{n-1}^h\|_V \leq \rho \|e_{n-1}^h\|_V.$$

By induction we have

$$(4.13) \qquad \|e_n^h\|_V \leq \rho^{n+1} \|u_h\|_V \leq \rho^{n+1} \frac{1}{\lambda} \|F\|_{V^*}.$$

The proof is complete. $\square$

An immediate corollary is the following global error estimate.

THEOREM 4.3. *Let $u(\omega)$ be solution of (2.1) and $\{U_n^h(\omega)\}_{n \geq 0}$ be generated by Algorithm 2. Then, there holds*

$$(4.14) \qquad \|u(\omega) - U_n^h(\omega)\|_V \leq C(h^\ell + \rho^{n+1}) \qquad \forall \omega \in \Omega.$$

Proof. The proof follows from Proposition 4.1, Theorem 4.2 and an application of the triangular inequality to the decomposition $u - U_n^h = (u - u_h) + (u_h - U_n^h)$. $\square$

5. Computational complexity and implementation strategies.

5.1. Linear solvers. Steps in Algorithm 2 solve linear systems with a coefficient matrix independent of the parameter ω, which is a highly appealing feature when a group of problems, associated to different ω's, needs to be solved. Since their discrete systems would share a common coefficient matrix, one can solve them simultaneously from a single system with multiple right-hand-side (RHS) vectors. Suppose J problems are considered corresponding to $\{\omega_1, \ldots, \omega_J\}$. For jth problem, at the algebraic level, Algorithm 2 finds $\mathbf{u}_n^j$ such that: for $n \geq 0$,

$$(5.1) \qquad \mathbf{A}_0 \mathbf{u}_n^j = \mathbf{b}_n^j = \begin{cases} \mathbf{f}^j & \text{if } n = 0, \\ \mathbf{f}^j - \mathbf{A}_1^j \mathbf{u}_{n-1}^j & \text{otherwise,} \end{cases}$$

where $\mathbf{A}_0$ is an $N \times N$ matrix. The set of systems can be recast into a matrix form:

$$(5.2) \qquad\qquad\qquad \mathbf{A}_0 \mathbf{U}_n = \mathbf{B}_n,$$

where $\mathbf{U}_n = [\mathbf{u}_n^1, \ldots, \mathbf{u}_n^J]$ is the solution matrix and $\mathbf{B}_n = [\mathbf{b}_n^1, \ldots, \mathbf{b}_n^J]$ contains RHS information from individual problems.

Based on the structure and size of the coefficient matrix $\mathbf{A}_0$, a direct solver or an iterative one can be used to find the solutions. Generally speaking, Algorithm 2 has several computational advantages: (i) $\mathbf{A}_0$ only needs to be accessed once for all the problems; when accessing or generating $\mathbf{A}_0$ represents a major bottleneck of a linear solver, this leads to a significant computational advantage [2]; (ii) A factorization of $\mathbf{A}_0$, if used, only needs to be implemented once in solving all the problems; (iii) The multiple RHS vectors in $\mathbf{B}_n$ lead to more efficient matrix-matrix products than matrix-vector products [6, Section 5.6]. Moreover, if a block Krylov subspace algorithm is used in solving (5.2), the multiple RHS vectors would enlarge the search space for minimizing residuals and thus could accelerate the convergence. We consider the direct solver in the paper and refer the readers to [16] and reference therein for discussions of block iterative solvers.

A direct solver contains the build phase and the solve phase, in which the former normally has higher complexity than the later. For a dense linear system, the factorization takes $\mathcal{O}(N^3)$ while the solve phase requires $\mathcal{O}(N^2)$. Thus, using Algorithm 2 is more efficient than solving problems separately as factorization is needed only once. Since $\mathbf{A}_0$ is sparse and SPD in the finite element discretization, fast direct solvers are available: using the nested dissection methods [10] or multi-frontal methods, the complexity for building the LU/Cholesky factorization is $\mathcal{O}(N^{3/2})$ in 2-D problem and $\mathcal{O}(N^2)$ in 3-D (see, for instance, [6, Section 9.3] and [5, Section 7.6]). Solving the resulting triangular systems has the complexity $\mathcal{O}(N \log N)$ in 2-D and $\mathcal{O}(N^{4/3})$ in 3-D. We next denote the cost of the build stage by $\mathcal{O}(N^p)$ and that of the solve stage by $\mathcal{O}(C_s)$, and analyze the total complexity of Algorithm 2 for solving J problems. Suppose the algorithm converges after $K - 1$ iterations, the total computation complexity is $\mathcal{O}(N^p + KJC_s)$. For which, one factorization of $\mathbf{A}_0$ costs $\mathcal{O}(N^p)$ and KJ times solves of two triangular systems cost $\mathcal{O}(KJC_s)$. But solving J problems separately has the complexity $\mathcal{O}(J(N^p + C_s))$. The ratio of the latter to the former provides a speedup factor

$$S_f = \frac{J(N^p + C_s)}{N^p + KJC_s} = \frac{1 + C_s N^{-p}}{J^{-1} + KC_s N^{-p}}.$$

Thus, given a finite K, for a fixed N, if J is big enough, the factor is on the order of $\frac{N^p}{KC_s}$ that scales as $\mathcal{O}(\sqrt{N}(\log N)^{-1}K^{-1})$ in 2-D and $\mathcal{O}(N^{\frac{2}{3}}K^{-1})$ in 3-D; while for a fixed J, as N becomes sufficiently large, the factor scales as J in both 2-D and 3-D.

5.2. Grouping. Given a group of problems, the convergence of the iterative algorithm could be slow if ρ is close to 1. In such a case, we propose to divide the problems into smaller groups to ensure a small ρ for each group. Motivated by the Centroidal Voronoi tessellations (CVT) method [22], we introduce the following algorithm.

Consider a set $W \subset \mathbb{R}^{\widehat{d}}$. A set $\{V_i\}_{i=1}^{n_c}$ is a tessellation of W if $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\cup_{i=1}^{n_c} V_i = W$. Let $|\cdot|$ denote the Euclidean norm on $\mathbb{R}^N$ and, for any v with magnitude greater than 0, define

$$r(x, v) = \frac{|x - v|}{|v|}.$$

Given a set of points $Z = \{z_i\}_{i=1}^{n_c}$, define the set V_i by

$$V_i = \{x \in W : r(x, z_i) \le r(x, z_j) \text{ for } j = 1, \dots, k, \ j \ne i\},$$

where the equality holds only for $i < j$. For any region $V \subset \mathbb{R}^{\widehat{d}}$, z^* denotes the center of the region. In a discrete setting, suppose there are n_s elements $\{x_i\}_{i=1}^{n_s} \in V$, we have

$$z^* = \frac{1}{n_s} \sum_{i=1}^{n_s} x_i.$$

Given a set of samples, we need to determine the regions V_i and centers z_i. Thus, we design an algorithm, presented in Algorithm 3, that uses iterations to find them. First, it initializes the regions (also referred to be clusters) and centers (also referred to be cluster generators). For each data point $x \in W$, the algorithm finds the nearest generator $z \in Z$. If that does not match the cluster to which the data point is currently assigned, the data point is "transferred" to the cluster associated with the nearest generator. When all data points have been considered, the cluster generators z are replaced by centers of the clustered data points. The process ends when there is no further "transfers" occurred.

Algorithm 3: Grouping algorithm

Input: Sample set W, number of samples n_s, number of centers n_c, maximum number of iterations $iter_{max}$.
Output: Centers Z, Regions $V_1, \dots, V_{n_c}$.
Initialize centers $Z = \{z_1, \dots, z_{n_c}\}$ uniformly spaced in the set;
Initialize regions $\{V_1, \dots, V_{n_c}\}$ associated to each z_j using the empty set, for
  $j = 1, \dots, n_c$;
for $k = 1 : iter_{\max}$ **do**
    for $i = 1 : n_s$ **do**
        Take i-th sample $x_i = W(i)$;
        Determine $z_j = \arg\min\limits_{v \in Z} r(x_i, v)$, where $r(x_i, v) = \frac{|x_i - v|}{|v|}$;
        Include x_i into the j-th region V_j controlled by z_j;
    end
    Update z_j by the center of V_j, for $j = 1, \dots, n_c$;
    if *no transfers occurred* **then**
      | Exit the loop
    end
end

---

6. Applications. In this section, we conduct three sets of numerical tests on three application problems. The first one solves a parameter-dependent diffusion problem, the second solves a random diffusion problem, and the third simulates a random convection-dominated convection-diffusion problem which is a randomized double-glazing problem [18]. Both 1-D and 2-D numerical test results are presented to demonstrate the efficiency of the proposed iterative method and to validate the theoretical convergence results. Particularly, the last two tests show that the proposed iterative method can successfully solve some strongly random diffusion and convection-diffusion problems which the multi-modes method fails to do.

6.1. Parameter-dependent diffusion problems. For a finite set of parameters $\{\omega_j\}_{j=1}^J$, let $a_j(x) = a(\omega_j, x)$, $f_j = f(\omega_j, x)$ and consider the following parameter-dependent diffusion problems:

$$-\nabla \cdot (a_j \nabla u_j) = f_j \qquad \text{in } D, \tag{6.1a}$$

$$u_j = 0 \qquad \text{on } \partial D, \tag{6.1b}$$

where D is a bounded Lipschitz domain in $\mathbb{R}^d$ for $d = 1, 2, 3$.

We assume the above PDE is uniformly elliptic, that is, there exist two constants $0 < \lambda < \Lambda$ such that the diffusion coefficient $\{a_j(x)\}_{j=1}^J$ satisfies

$$\lambda \leq a_j(x) \leq \Lambda \qquad \forall x \in D, \, j = 1, 2, \cdots, J. \tag{6.2}$$

In addition, assume that $a_j(x) = a_0(x) + \eta_j(x)$ with $\eta_j(x) = \eta(\omega_j, x)$ and $a_0(x)$ satisfies

   (i) Uniform ellipticity: $0 < \underline{a}_0 \leq a_0(x) \leq \overline{a}_0$.
   (ii) Relative dominance: there exists a number $\rho \in (0, 1)$ such that

$$\frac{\|a_j - a_0\|_{L^\infty}}{\underline{a}_0} = \frac{\|\eta_j\|_{L^\infty}}{\underline{a}_0} < \rho \qquad \forall j = 1, \cdots, J. \tag{6.3}$$

There are a couple of options to choose $a_0(x)$ which are given below (cf. [20]).
   (a) $a_0(x)$ is chosen as the arithmetic average of $\{a_j(x)\}_{j=1}^J$, that is,

$$a_0(x) := \frac{1}{J}\Big(a_1(x) + a_2(x) + \cdots + a_J(x)\Big). \tag{6.4}$$

   (b) $a_0(x)$ is chosen as the largest value of $\{a_j(x)\}_{j=1}^J$, that is,

$$a_0(x) := \max_{1 \leq j \leq J} a_j(x). \tag{6.5}$$

To fit the abstract framework, we set $V = H_0^1(D)$ and define the bilinear forms A, A_0, A_1 and the linear functional F respectively as

$$A(\omega; u, v) = (a(\omega, \cdot)\nabla u, \nabla v), \qquad F(\omega; v) = (f(\omega, \cdot), v),$$
$$A_0(u, v) = (a_0 \nabla u, \nabla v), \qquad A_1(\omega; u, v) = (\eta(\omega, \cdot)\nabla u, \nabla v).$$

It is easy to verify that A, A_0, A_1 and F satisfy the convergence criteria laid out in Sections 3 and 4. Hence, Theorems 3.1, 4.2 and 4.3 apply to this problem with $\ell = r$.

Two test cases are considered in this section: one is a 1-D diffusion with an analytic solution, the other is a 2-D diffusion with more degrees of freedom but no analytic solution. We use the former to illustrate the theoretical results and the latter for checking the numerical efficiency. To investigate the numerical performance of the iterative algorithm, we consider two metrics:

$$\mathcal{E}_j = \|u_j - u_{j,n}^h\|_{H^1} \quad \text{and} \quad \mathcal{E}_j^h = \|u_j^h - u_{j,n}^h\|_{H^1},$$

where u_j is the exact solution, u_j^h is the finite element solution from individual simulations and $u_{j,n}^h$ is the finite element solution at n-th iteration of the iterative algorithm.

Test 1. Consider a 1-D diffusion problem (with homogeneous boundary condition) on $D = [0, 1]$ with the diffusion coefficient

$$a_j(x) = 1 + x + \epsilon_j \sin(x).$$

The exact solution is given by $u(x) = x(x-1) + 0.5\sin(20\pi x) + \epsilon_j \sin(40\pi x)$, and the source term is determined by plugging the exact solution in (6.1). For investigating the iterative algorithm, we consider $J = 5$ problems with the choices of $\epsilon_j \in \{0.1035, 0.0727, -0.0303, 0.0294, -0.0787\}$.

For the spatial discretization, the quadratic conforming finite elements ($\ell = 2$) are used on a uniform mesh with the size h. In the iterative algorithm, we set the stopping criterion to be $\max_j \|u_{j,n+1}^h - u_{j,n}^h\|_{H^1} < \mathsf{tol} = 10^{-4}$, that is, the maximum H^1 norm of the differences between numerical solutions of adjacent iterations is less than tol.

Firstly, we choose $a_0(x) = 1 + x + \bar{\epsilon}\sin(x)$ with $\bar{\epsilon} := \frac{1}{5}\sum_{j=1}^5 \epsilon_j = 0.0193$. The iterative algorithm takes 4 iterations to complete for meshes with different resolutions. The numerical errors $\mathcal{E}_j$, for $j = 1, \ldots, 5$, are listed in Table 6.1, which shows for each of the five problems, the approximation error decays at the second order as mesh is uniformly refined. The convergence history of $\mathcal{E}_j^h$ for the five problems is shown in Figure 6.1 by taking the $h = 2^{-10}$ case for example. Applying regressions on the results shows $\mathcal{E}_1^h \sim \mathcal{O}(0.032^{n+1})$ in problem 1, $\mathcal{E}_2^h \sim \mathcal{O}(0.021^{n+1})$ in problem 2, $\mathcal{E}_3^h \sim \mathcal{O}(0.019^{n+1})$ in problem 3, $\mathcal{E}_4^h \sim \mathcal{O}(0.0039^{n+1})$ in problem 4, and $\mathcal{E}_5^h \sim \mathcal{O}(0.038^{n+1})$ in problem 5. The ratios $\rho_j = \frac{\|a_j - a_0\|_{L^\infty}}{a_0}$ for these five problems are 0.071, 0.045, 0.042, 0.009, and 0.083, respectively, which are close and proportional to the regression rates. We found a better ratio that matches the regression analysis result can be defined by $\widehat{\rho}_j = \left\|\frac{a_j - a_0}{a_0}\right\|_{L^\infty}$, whose values are respectively 0.035, 0.022, 0.021, 0.004 and 0.041 for these problems. Further analysis in this aspect will be performed in a future work.

h	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_3$	$\mathcal{E}_4$	$\mathcal{E}_5$
$\frac{1}{2^7}$	3.82×10^{-1}	3.03×10^{-1}	2.21×10^{-1}	2.19×10^{-1}	3.18×10^{-1}
$\frac{1}{2^8}$	9.62×10^{-2}	7.63×10^{-2}	5.54×10^{-2}	5.50×10^{-2}	8.00×10^{-2}
$\frac{1}{2^9}$	2.41×10^{-2}	1.91×10^{-2}	1.39×10^{-2}	1.38×10^{-2}	2.00×10^{-2}
$\frac{1}{2^{10}}$	6.03×10^{-3}	4.78×10^{-3}	3.46×10^{-3}	3.44×10^{-3}	5.01×10^{-3}

TABLE 6.1
Errors $\mathcal{E}_j$, for $j = 1, \ldots, 5$, of the iterative algorithm solutions at different h.
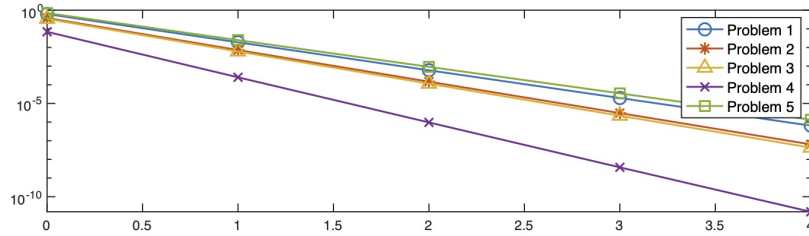
FIG. 6.1. *Evolution of $\mathcal{E}_j^h$, for $j = 1, \ldots, 5$, during the iteration.*

Secondly, we choose $a_0 = a_\infty := \max_j \max_{x \in D} |a_j(x)| = 2.0871$ while keeping the same computational setting as the previous case. The iterative algorithm achieves the same errors listed in Table 6.1, which is not surprising because the same stopping criterion is used in both tests. However, the total number of iterations increases to

16 in this case. The evolution of $\mathcal{E}_j^h$ for the five problems during the iteration when $h = 2^{-10}$ is plotted in Figure 6.2. Regressions on the data shows $\mathcal{E}_j^h \sim \mathcal{O}(0.49^{n+1})$ in these 5 problems. Note that the ratios $\rho_j = \widehat{\rho}_j$ in this case because a_0 is a constant. For the five problems, the ratios are about 0.52 that are close to the regression results.
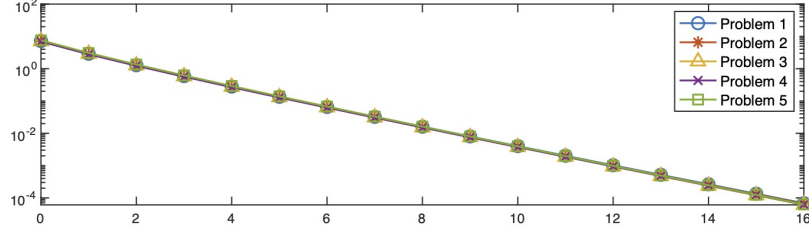

FIG. 6.2. *Evolution of $\mathcal{E}_j^h$, for $j = 1, \ldots, 5$, during the iteration.*

It is observed that the numerical results match the theoretical analysis in Theorems 4.2 - 4.3. In particular, $\mathcal{E}_j$ is dominated by the finite element discretization error, that is of the order $\mathcal{O}(h^\ell)$ with $\ell = 2$; and $\mathcal{E}_j^h$ is of the order $\mathcal{O}(\rho_j^{n+1})$, for $j = 1, \ldots, 5$.

Test 2. We consider a diffusion problem on the domain $D = [-1, 1]^2$. Let D_1 be a disk centered at the origin of radius 0.5 and $D_0 = D\backslash\overline{D}_1$. The conductivity $a(x)$ is a piecewise constant on D:

$$a(x)|_{D_1} = \mu^{[1]} \quad \text{and} \quad a(x)|_{D_0} = 1.$$

The problem has zero forcing and is associated to a Dirichlet boundary condition on the top edge and Neumann boundary conditions on the other sides. In particular,

$$u = 0 \text{ on } \Gamma_{top} = [-1, 1] \times \{1\},$$
$$a\nabla u \cdot n = 0 \text{ on } \Gamma_{side} = \{\pm 1\} \times (-1, 1),$$
$$a\nabla u \cdot n = \mu^{[2]} \text{ on } \Gamma_{bottom} = [-1, 1] \times \{-1\}.$$

Denote the parameter vector by $\boldsymbol{\mu} = (\mu^{[1]}, \mu^{[2]})$, whose ranges is specified by

$$\Omega := \{\boldsymbol{\mu} \in \mathbb{R}^2;\ 0.1 \leq \mu^{[1]} \leq 10,\ -1 \leq \mu^{[2]} \leq 1\}.$$

In this case, the bilinear forms A, A_0, A_1 and the linear functional F have the following form:

$$A(\boldsymbol{\mu}; u, v) = \left(\mu^{[1]}\nabla u, \nabla v\right)_{D_1} + \left(\nabla u, \nabla v\right)_{D_0}, \quad F(\boldsymbol{\mu}; v) = \left(f(\boldsymbol{\mu}), v\right) + \langle\mu^{[2]}, v\rangle_{\Gamma_{bottom}},$$
$$A_0(u, v) = \left(\mu_0^{[1]}\nabla u, \nabla v\right)_{D_1} + \left(\nabla u, \nabla v\right)_{D_0}, \quad A_1(\boldsymbol{\mu}; u, v) = \left((\mu^{[1]} - \mu_0^{[1]})\nabla u, \nabla v\right)_{D_1},$$

where we use $a_0|_{D_1} := \mu_0^{[1]}$ and $a_0|_{D_0} := 1$. Since these forms have affine parameter dependence, the assembly of the associated matrices and vectors can be greatly simplified in computation.

We use the quadratic conforming finite element method for spatial discretization on a triangulation of D with the mesh size h. For instance, the triangulation of $h = 1/8$ is shown in Figure 6.3 (left), which contains 496 elements in total. In the iterative algorithm, the stopping criterion $\mathsf{tol} = 10^{-4}$ is selected.
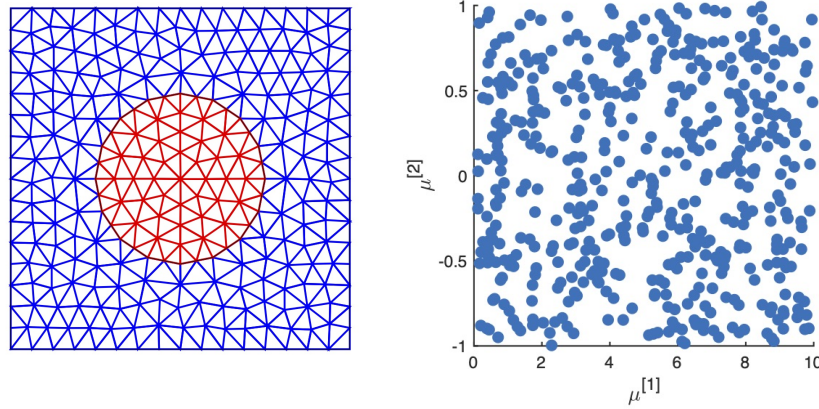
FIG. 6.3. *(Left) A finite element mesh with 496 elements; (Right) $\boldsymbol{\mu}$-samples.*

There are n_s $\boldsymbol{\mu}$-samples $\{\boldsymbol{\mu}_j = (\mu_j^{[1]}, \mu_j^{[2]})\}_{j=1}^{n_s}$ randomly chosen in the parameter domain, e.g., $n_s = 500$ randomly selected samples are shown in Figure 6.3 (right). Note that only $\mu_j^{[1]}$ appears in the bilinear form, the ratio ρ_j only depends on $\mu_j^{[1]}$. In such a case, choosing $\mu_0^{[1]} := \max_j \mu_j^{[1]}$ requires many iterations as shown in previous test, but choosing $\mu_0^{[1]} := \frac{1}{J} \sum_{j=1}^{n_s} \mu_j^{[1]}$ also will lead to a big ρ_j for some problems and require many iterations as well. Therefore, we first divide the sample set into multiple subgroups using Algorithm 3 and then apply the iterative algorithm to each of them.

We use a mesh of size $h = \frac{1}{32}$ for spatial discretization, which contains 8,136 elements and 16,529 nodes. For the same set including $n_s = 500$ samples, Algorithm 3 is fed with $W := \{\mu_1^{[1]}, \ldots, \mu_{n_s}^{[1]}\}$ and finds $n_c = 10$ regions and associated centers. The evolution of centers during the process is shown in Figure 6.4 (left), which shows the sample points stop transferring after 78 steps. Each sample from the set is assigned to a region V_k with the center z_k, for $k = 1, \ldots n_c$. The ratio $r(\mu_i^{[1]}, z_k)$ for each $\boldsymbol{\mu}_i$ is shown in Figure 6.4 (right). It is seen that all these ratios are below 0.3. The iterative algorithm is sequentially performed in the n_c groups with $a_0|_{D_1} := z_k$ for each group. The entire simulation is completed in 29.48 seconds and the maximum number of iterations for each group is 5. The detailed information for all the groups is listed in Table 6.2, where ρ represents the maximum value of $r(\mu_i^{[1]}, z_j)$ for elements in j-th group. The maximum errors between the iterative solution and the finite element solution, $\max_j \mathcal{E}_j^h$, in each group are plotted in Figure 6.5.

Next, we vary the number of samples and compare the iterative algorithm with the individual simulations in terms of wall-clock times. Since the exact solution is unknown, we measure the maximum difference in H^1 norm between iterative algorithm solutions and individual solutions. The results are listed in Table 6.3, where T_{it} represents the time elapsed for integration in the iterative algorithm and T_{ind} is that for integrating individual systems. It is observed that as the size of sample set increases, the efficiency of iterative algorithm improves. When $n_s = 2500$, the iterative algorithm saves about 50% simulation time.

Finally, we fix $n_s = 2500$ while doubling the number of centers n_c from 5 to 160. The wall-clock times T_{it} are listed in Table 6.4, which shows the simulation time first decreases then increases. The least computational time is achieved when $n_c = 80$,
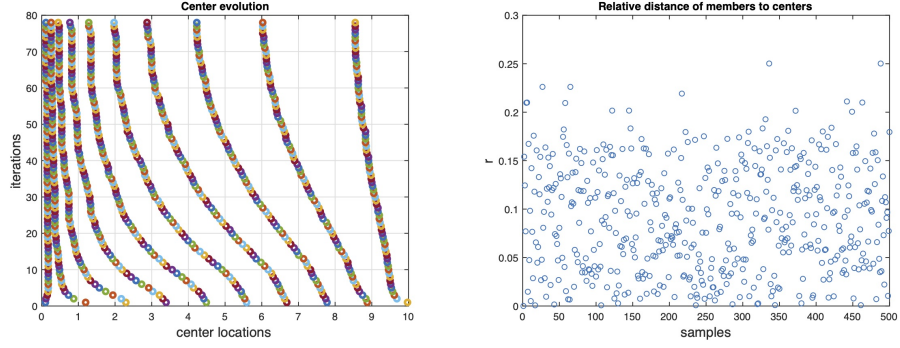
FIG. 6.4. *(Left) Evolution of centers of 10 subgroups; (Right) $r(\mu_i^{[1]}, z_j)$ for each $\boldsymbol{\mu}_i \in W$ belongs to V_j and z_j is the center of V_j.*

| group	size	$\mu^{[1]}$-region	center	ρ	iterations	max $\mathcal{E}_j^h$	time
1	4	[0.10, 0.13]	0.12	0.12	3	2.15×10^{-7}	0.49
2	15	[0.19, 0.32]	0.26	0.25	4	3.92×10^{-6}	1.06
3	14	[0.36, 0.54]	0.45	0.20	4	4.99×10^{-6}	1.05
4	17	[0.59, 0.94]	0.77	0.23	5	3.81×10^{-6}	1.37
5	33	[1.01, 1.55]	1.28	0.21	5	5.48×10^{-6}	2.18
6	32	[1.61, 2.33]	1.97	0.18	5	3.05×10^{-6}	2.12
7	50	[2.36, 3.39]	2.87	0.18	5	4.55×10^{-6}	3.07
8	68	[3.49, 4.97]	4.23	0.17	5	5.08×10^{-6}	3.90
9	111	[4.99, 7.06]	6.03	0.17	5	2.78×10^{-6}	6.15
10	156	[7.12, 9.97]	8.55	0.17	5	3.72×10^{-6}	8.09

TABLE 6.2

Information of subgroups: size, region, center and maximum ρ; performance of the iterative algorithm including number of iterations, maximum errors and wall-clock time for simulations.

n_s	T_{it}	T_{ind}	$\max_j \mathcal{E}_j^h$
100	8.09	8.74	7.2×10^{-6}
500	29.48	43.14	5.48×10^{-6}
2500	116.10	233.23	1.27×10^{-6}

TABLE 6.3

Comparison of the iterative algorithm with individual simulations at $h = 1/32$: wall-clock times and maximum values of $\mathcal{E}_j^h$ for $j = 1, \ldots n_s$.

which saves over 60% simulation time compared to the individual simulations. It is also observed that as the number of groups increases, the maximum difference between iterative algorithm and individual simulation solutions, $\max_j \mathcal{E}_j^h$, decreases, which is because ρ shrinks as the problems are divided into more groups.
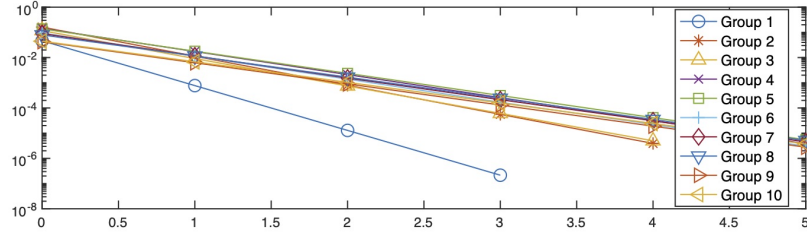
FIG. 6.5. *Evolution of* $\max_j \mathcal{E}_j^h$ *for each group during the iteration.*

n_c	5	10	20	40	80	160
T_{it}	187.5	116.1	102.64	93.01	89.69	111.75
$\max_j \mathcal{E}_j^h$	0.0000551	0.0000127	0.000005	0.00000327	0.00000328	0.00000167

TABLE 6.4

Wall-clock times and maximum values of $\mathcal{E}_j^h$ *in the iterative algorithm for* 2500 *samples that are divided into* n_c *groups.*

6.2. Random diffusion equations. In this subsection we consider the following random diffusion test problem:

$$-\nabla \cdot \big(a(\omega, \cdot)\nabla u\big) = f(\omega, \cdot) \qquad \text{in } D, \quad \mathbb{P}\text{-a.s.} \tag{6.6a}$$

$$u = 0 \qquad \text{on } \partial D, \quad \mathbb{P}\text{-a.s.} \tag{6.6b}$$

Where D is a bounded Lipschitz domain in $\mathbb{R}^d$ for $d = 1, 2, 3$. $a(\omega, x)$ and $f(\omega, x)$ are random fields on the probability space $(\Omega, \mathbb{P}, \mathcal{F})$ and they are assume to have continuous and bounded covariance function. $\mathbb{E}[\cdot]$ denotes the expectation operator. In addition, a satisfies $\mathbb{P}$-a.s. the following condition:

$$0 < \lambda \le a(\omega, x) \le \Lambda \qquad \forall x \in D. \tag{6.7}$$

Moreover, we assume that $\mathbb{E}[\|f\|_{L^2(D)}] < \infty$. Below we propose two approaches for utilizing the abstract framework for problem (6.6). The first approach mimics the multi-modes method of [7] while the second one mimics the ensemble method of [20].

6.2.1. Approach #1. Suppose that there exists following decomposition of a:

$$a(\omega, x) = a_0(x) + \eta(\omega, x) \qquad \text{a.s. } \forall x \in D, \tag{6.8}$$

where a_0 is independent of ω and also satisfies
  (i) *Uniform ellipticity:* $0 < \underline{a_0} \le a_0(x) \le \bar{a}_0$ for all $x \in D$.
  (ii) *Relative dominance:* there exists a number $\rho \in (0, 1)$ such that

$$\mathbb{P}\big\{\omega \in \Omega; \|\eta(\omega, \cdot)\|_{L^\infty(D)} \le \rho \underline{a_0}\big\} = 1. \tag{6.9}$$

To apply the abstract framework, we define

$$A(\omega; u, v) = \big(a(\omega, \cdot)\nabla u, \nabla v\big), \qquad F(\omega; v) = \big(f(\omega, \cdot), v\big),$$
$$A_0(u, v) = \big(a_0 \nabla u, \nabla v\big), \qquad A_1(\omega; u, v) = \big(\eta(\omega, \cdot)\nabla u, \nabla v\big).$$

It is easy to verify that A, A_0, A_1 and F satisfy $\mathbb{P}$-a.s. the convergent criteria laid out in Sections 3 and 4. Hence, Theorems 3.1, 4.2 and 4.3 apply to this problem with $\ell = r$. In particular, there holds

$$(6.10) \qquad \mathbb{E}\big[\|u - U_n^h\|_{H^1}\big] \leq C\big(h^r + \rho^{n+1}\big).$$

It should be noted that $C > 0$ is independent of ω because $\mathbb{E}[\|f\|_{L^2(D)}] < \infty$.

We remark that in this first approach we do not specify how to discretize the stochastic variable ω. However, since the abstract framework suits best with the random sampling method, in practice, the Monte Carlo method would be used to discretize the right-hand sides in Step 1 and 2 of Algorithm 2.

Finally, we note that the choice of a_0 depends on the structure of a (cf. [7]).

To verify the convergence rates in (6.10) of this approach, we consider the following random coefficient boundary value problem [7] for our numerical tests:

$$(6.11) \qquad -\frac{d}{dx}\left(\big(1 + \varepsilon X(\omega)\big)\frac{du(\omega, x)}{dx}\right) = X(\omega), \qquad 0 < x < 1,$$
$$u(\omega, 0) = 0, \quad u(\omega, 1) = 0,$$

where $X(\omega)$ is a uniformly distributed random variable defined in a probability space $(\Omega, \mathcal{B}, \mathbb{P})$, where the sample space $\Omega = [0, 1]$, $\mathcal{B}$ denotes the σ-algebra of the Borel sets and $\mathbb{P}$ denotes the Lebesgue probability measure. In addition, we assume that $\varepsilon > 0$. The true solution of (6.11) is given by

$$(6.12) \qquad u(\omega, x) = \frac{X(\omega)}{2(1 + \varepsilon X(\omega))}(x - x^2).$$

We have that $a(\omega, x) = 1 + \varepsilon X(\omega)$ satisfies the elliptic condition with $\lambda = 1, \Lambda = 1 + \varepsilon$. In addition, $a(\omega, x) = a_0(x) + \eta(\omega, x)$, where $a_0(x) = 1$ and $\eta(\omega, x) = \varepsilon X(\omega)$ satisfies the conditions (i) with $\underline{a}_0 = 1$ and (ii) with $\rho = \varepsilon$. In order to fit the abstract framework, in this example, we define $V = H_0^1(0, 1)$ and

$$A(\omega; u, v) = \big((1 + \varepsilon X(\omega))u', v'\big), \qquad F(\omega; v) = \big(X(\omega), v\big),$$
$$A_0(u, v) = \big(u', v'\big), \qquad\qquad A_1(\omega; u, v) = \big(\varepsilon X(\omega)u', v'\big).$$

Let V_1^h denote the standard linear finite element subspace of V which is used for the spatial discretization of (6.11). Let $\{U_n^h\}_{n \geq 0}$ be the approximate solution from Algorithm 2, where n is the number of iterations. Let

$$(6.13) \qquad \mathcal{E}_{L^2}^n := \mathbb{E}\big[\|u - U_n^h\|_{L^2}\big], \quad \mathcal{E}_{H^1}^n := \mathbb{E}\big[\|u - U_n^h\|_{H^1}\big].$$

As mentioned above, we use the Monte Carlo method to discretize the stochastic variable ω on the right-hand side in Step 1 and 2 of Algorithm 2. The number of the Monte Carlo samples is chosen to be $J = 10^4$. To test the validity and accuracy of the proposed iterative method, we set $h = 0.01$ for spatial discretization and then vary the number of iterations n. Table 6.5 displays the H^1-norm errors of the iterative solutions for different values of n and ε. As expected, the computed solutions have smaller errors for smaller $\rho = \varepsilon$, and the method converges as long as $\varepsilon < 1$ which is predicted by our convergence theorem. In addition, our numerical tests show that the method ceases to converge for $\rho = \varepsilon > 1$, this indicates that our convergence result is sharp.

ρ	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
0.4	0.093	0.0033	0.0016	0.00123	0.00115	0.00114
0.6	0.0184	0.0089	0.0047	0.0027	0.0018	0.0014
0.8	0.0295	0.0187	0.0124	0.0086	0.0062	0.0046
0.9	0.0356	0.0253	0.188	0.0145	0.0115	0.0093

TABLE 6.5

Approach #1: Errors $\mathcal{E}_{H^1}^n$ of the computed solutions with different ρ and n.

h	$\mathcal{E}_{H^1}^n$	Order	$\mathcal{E}_{L^2}^n$	Order
0.2	2.43×10^{-2}		1.5000×10^{-3}	
0.1	1.29×10^{-2}	0.92	4.0659×10^{-4}	1.88
0.05	6.60×10^{-3}	0.97	1.0443×10^{-4}	1.96
0.025	3.30×10^{-3}	1.00	2.6450×10^{-5}	1.98

TABLE 6.6

Errors $\mathcal{E}_{H^1}^n$ and $\mathcal{E}_{L^2}^n$, for $n=10$, of the iterative algorithm solutions at different h.

To test the convergence orders for the spatial discretization, we fix $\rho = \varepsilon = 0.1$ and the number of iterations $n = 10$, then vary the mesh size h. Table 6.6 shows the H^1- and L^2-errors of the computed solution by Algorithm 2. We observe that an $O(h)$ order convergent rate in the H^1-norm and an $O(h^2)$ order rate in the L^2-norm are obtained for the numerical solution U_n^h which is consistent with the theoretical error estimate in (6.10).

Next, we verify the dependence of the errors on the parameter $\rho = \varepsilon$. To the end, we fix $\rho = 0.4, h = 0.01$ and then vary the number of iterations n. We also use the stopping criteria $\mathbb{E}\big[\|U_{n+1}^h - U_n^h\|_{H^1}\big] < \texttt{tol} := 10^{-4}$. The H^1- and L^2-norm errors of the computed solutions are shown in Figure 6.6. It is clear that the errors are decreasing as the number of iterations increases and the convergence order $O(\rho^{n+1})$ is indeed observed as predicted in (6.10). In addition, Figure 6.6 also shows that only 6 iterations is needed to trigger the stopping criterion.

Furthermore, we plot the errors of the first five Monte Carlo samples in Figure 6.7 to show that the errors for different samples are different and they are also different from the expected value which is presented in Figure 6.6, but they all are decreasing as n increases provided that ε is small.

Finally, to check the dependence of the iterative method on the parameter ε, we fix $n = 10$, $h = 0.01$ and then consider $\varepsilon = 2, 2.5, 3, 3.5$, which all are larger than 1. We note that in these cases, the relative dominant condition (ii) is violated, hence, our convergence results do not apply anymore. Figure 6.8 shows evidently that the H^1- and L^2-norm errors increase significantly as ε becomes larger, which clearly indicates that the iterative method may diverge when $\varepsilon > 1$ and also shows that the relative dominant condition (ii) is sharp.

6.2.2. Approach #2. In this subsection, we present a different approach to utilize the proposed iterative method for solving the same random diffusion problem (6.6) based on the Monte Carlo finite element discretization. To solve (6.6) by the Monte Carlo method, let $\{\omega_j\}_{j=1}^J$ be J samples from the sample space Ω, we then
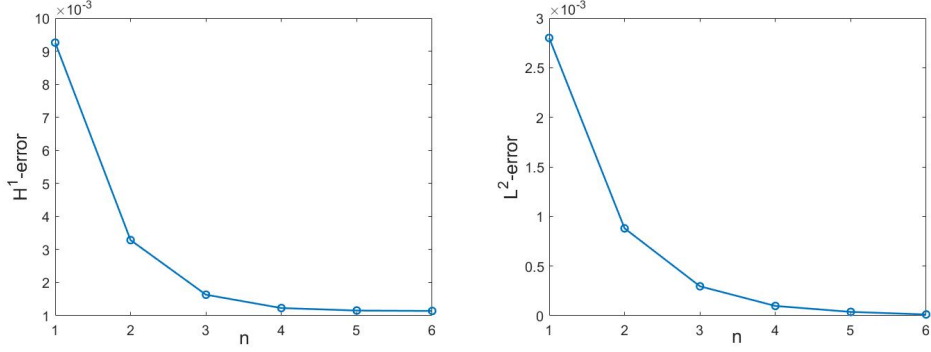
FIG. 6.6. *Approach #1: Errors $\mathcal{E}_{H^1}^n$ (left) and $\mathcal{E}_{L^2}^n$ (right) for $\rho = \varepsilon = 0.4$, $h = 0.01$,* `tol` $= 10^{-4}$ *and different n.*
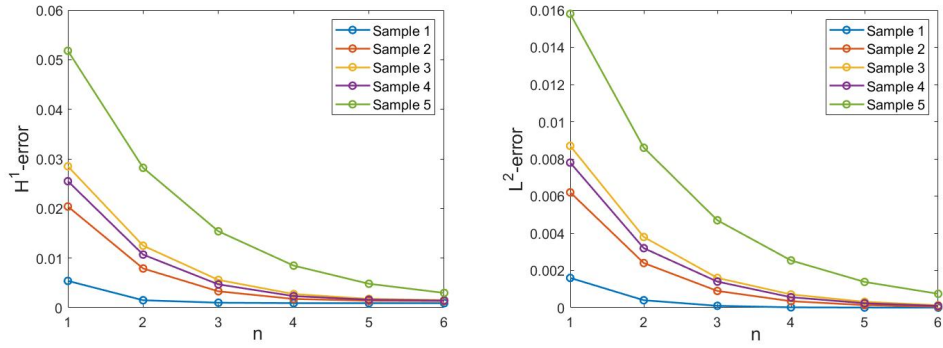

FIG. 6.7. *Approach #1: Errors $\mathcal{E}_{H^1}^n$ (left) and $\mathcal{E}_{L^2}^n$ (right) decay for the first 5 Monte Carlo samples with $\rho = \varepsilon = 0.4$, $h = 0.01$ and different n.*

consider (deterministic) problems for $j = 1, 2, \cdots, J$

(6.14a) $$-\nabla \cdot \big(a(\omega_j, \cdot)\nabla u\big) = f(\omega_j, \cdot) \qquad \text{in } D,$$
(6.14b) $$u = 0 \qquad \text{on } \partial D.$$

Hence, the Monte Carlo method converts the random diffusion problem (6.6) into the parameter-dependent problem (6.14) which is of the same type as problem (6.1). As a result, the method described in Section 6.1 readily applies to problem (6.14). The only extra step is to bound the expected value of the error as follows:

(6.15) $$\big\|\mathbb{E}[u] - \mathbb{E}_{mc}[U_n^h]\big\|_V \le \big\|\mathbb{E}[u] - \mathbb{E}[U_n^h]\big\|_V + \big\|\mathbb{E}[U_n^h] - \mathbb{E}_{mc}[U_n^h]\big\|_V$$
$$\le C\big(\rho^{n+1} + h^r\big) + CJ^{-\frac{1}{2}},$$

where $\mathbb{E}_{mc}[\cdot]$ denotes the Monte Carlo approximation of the expected value. In addition, unlike Approach #1 above, which is equivalent to the multimodes method of [7] and imposes the restrictive condition $0 < \varepsilon < 1$ for convergence because $\rho = \epsilon$, we like to show below that using Approach #2 the iterative method also converges for $\varepsilon > 1$ because we can have $\rho < 1$ even $\varepsilon > 1$ in this approach.

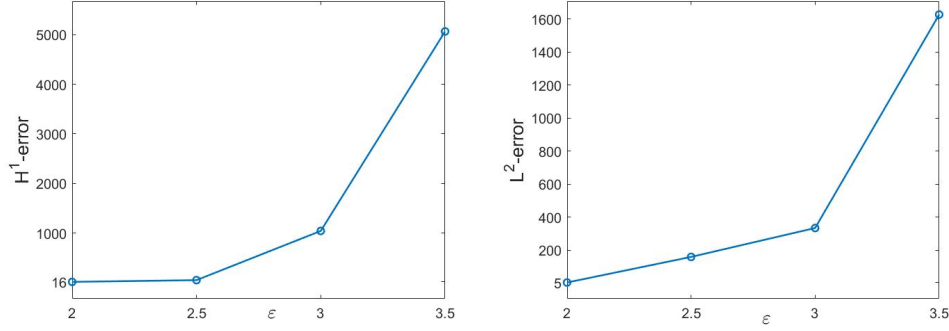We reconsider the random diffusion test problem (6.11) whose exact solution is

FIG. 6.8. *Approach #1: Errors $\mathcal{E}_{H^1}^n$ (left) and $\mathcal{E}_{L^2}^n$ (right) for different $\varepsilon > 1$ and $n = 10$.*

given by (6.12). It is easy to check that

$$\mathbb{E}[u] = \frac{1}{2}\left(\frac{1}{\varepsilon} - \frac{1}{\varepsilon^2}\ln(1+\varepsilon)\right)(x - x^2).$$

The parameter-dependent problem (6.14) now becomes

$$-\frac{d}{dx}\left(\left(1 + \varepsilon X(\omega_j)\right)\frac{du(\omega_j, x)}{dx}\right) = X(\omega_j), \qquad 0 < x < 1,$$
$$u(\omega_j, 0) = 0, \quad u(\omega_j, 1) = 0,$$

for $j = 1, \cdots, J$.

To fit the setup of Section 6.1, we also define the following bilinear forms:

$$A(\omega_j; u, v) = \left(a(\omega_j, \cdot)u', v'\right), \qquad F(\omega_j; v) = \left(X(\omega_j), v\right),$$
$$A_0(u, v) = \left(a_0\, u', v'\right), \qquad\qquad A_1(\omega_j; u, v) = \left(\eta(\omega_j, \cdot)u', v'\right),$$

where $a(\omega_j, x) = 1 + \varepsilon X(\omega_j)$; a_0 can be chosen as in Section 6.1. For example, we can take $a_0(x) = \max\limits_{1 \le j \le J} a(\omega_j, x)$ or $a_0(x) = \mathbb{E}_{mc}[a(\cdot, x)]$. Then, $\eta(\omega_j, x) = a(\omega_j, x) - a_0(x)$ will automatically satisfy the relative dominant condition (ii) with

$$\rho = \frac{\max\limits_{1 \le j \le J}\max\limits_{x \in D}|\eta(\omega_j, x)|}{\min\limits_{x \in D} a_0(x)} < 1.$$

We also introduce the following error functions:

$$\mathcal{E}_{L^2}^n = \|\mathbb{E}[u] - \mathbb{E}_{mc}[U_n^h]\|_{L^2}, \quad \mathcal{E}_{H^1}^n = \|\mathbb{E}[u] - \mathbb{E}_{mc}[U_n^h]\|_{H^1}.$$

We now want to verify the convergence rate given in (6.15) for the iterative method. Notice that there are three terms in the error estimate (6.15). Since the error (i.e., the term $CJ^{-\frac{1}{2}}$) due to the Monte Carlo method is standard and we omit it here. In order to neglect this Monte Carlo error, we choose a large sample number $J = 10^6$ in all the numerical tests below.

To verify the finite element method error term $O(h)$, we consider $\varepsilon = 2.0$ and fix $n = 10$ then choose different values of the mesh size h. It is easy to check that $\rho =$

0.5003 when $a_0(x) = \mathbb{E}_{mc}[a(\cdot,x)]$ and $\rho = 0.6667$ when $a_0(x) = \max_{1 \leq j \leq J} a(\omega_j, x)$. In both cases, $\rho < 1$, which satisfies the relative dominant assumption in the convergence theorem. Table 6.7 displays the H^1- and L^2-norm errors for the first choice of a_0 and Table 6.8 shows the errors for the second choice of a_0. We observe a convergence rate $O(h)$ for the H^1-norm errors in both cases which is consistent with our error estimate in (6.15). In addition, we observe that the L^2-norm errors exhibit an $O(h^2)$ order of the convergence.

h	$\mathcal{E}_{H^1}^n$	Order	$\mathcal{E}_{L^2}^n$	Order
0.2	1.17×10^{-2}		7.2850×10^{-4}	
0.1	6.20×10^{-3}	0.93	1.8724×10^{-4}	1.96
0.05	3.20×10^{-3}	0.96	4.2524×10^{-5}	2.14
0.025	1.60×10^{-3}	1.00	6.9804×10^{-6}	2.61

TABLE 6.7

Approach #2: Errors $\mathcal{E}_{H^1}^n$ and $\mathcal{E}_{L^2}^n$, for $\varepsilon = 2.0$, $a_0 = \mathbb{E}_{mc}[a] = 2.0011$, $\rho = 0.5003$, $n = 10$ and different h.

h	$\mathcal{E}_{H^1}^n$	Order	$\mathcal{E}_{L^2}^n$	Order
0.2	1.17×10^{-2}		7.3323×10^{-4}	
0.1	6.20×10^{-3}	0.93	1.9227×10^{-4}	1.93
0.05	3.20×10^{-3}	0.96	4.7262×10^{-5}	2.02
0.025	1.60×10^{-3}	1.00	9.9898×10^{-6}	2.24

TABLE 6.8

Approach #2: Errors $\mathcal{E}_{H^1}^n$ and $\mathcal{E}_{L^2}^n$, for $\varepsilon = 2.0$, $a_0(x) = \max_{1 \leq j \leq J} a(\omega_j, x) = 3.0$, $\rho = 0.6667$, $n = 10$ and different h.

To verify the iteration error term $O(\rho^{n+1})$, we fix $h = 0.01$, $\varepsilon = 2.0$ and also consider two cases of a_0 as in the previous test. A stopping criteria $\|\mathbb{E}_{mc}[U_{n+1}^h - U_n^h]\|_{H^1} < \texttt{tol} := 10^{-4}$ is used to terminate the iteration. Figure 6.9 displays the H^1- and L^2-norm errors of the computed solutions. We observe that the errors are decreasing as the number of iterations increases. Figure 6.9 also shows that only 6 iterations are needed to trigger the stopping criterion.

Finally, we plot the errors of the first five Monte Carlo samples in Figure 6.10 for $\varepsilon = 2.0$, $a_0(x) = \mathbb{E}_{mc}[a(\cdot,x)]$, $h = 0.01$ and various n. As expected, the errors are different for different samples and they all are also different from the expected value in Figure 6.9, but they all are decreasing as the number of iterations n increases.

6.3. 1-D random convection-diffusion problems. In this subsection, we consider the following 1-D randomized double-glazing problem [18]:

$$(6.16) \qquad -\frac{d}{dx}\left(\left(1 + \varepsilon X(\omega)\right)\frac{du(\omega,x)}{dx}\right) + 100\left(1 + \varepsilon X(\omega)\right)\frac{du(\omega,x)}{dx} = f(\omega,x),$$
$$u(\omega,0) = 0, \qquad u(\omega,1) = 0,$$

where $0 \leq x \leq 1$, $\varepsilon > 0$ and $X(\omega)$ is a uniformly distributed random variable on a probability space $(\Omega, \mathcal{B}, \mathbb{P})$ with $\mathcal{B}$ is the Borel σ-algebra. In addition, we choose
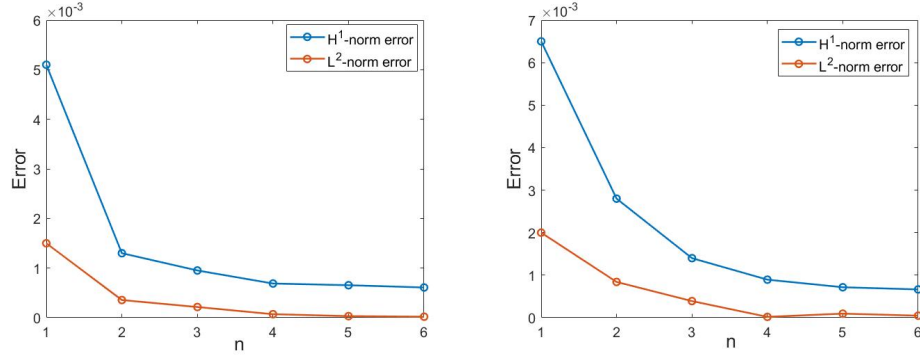
FIG. 6.9. *Approach #2: Errors $\mathcal{E}_{H^1}^n$ and $\mathcal{E}_{L^2}^n$ with $a_0(x) = \mathbb{E}_{mc}[a(\cdot, x)] = 2.0011$ (left) and $a_0(x) = \max_{1 \le j \le J} a(\omega_j, x) = 3.0$ (right) for $h = 0.01, \varepsilon = 2.0,$ tol $= 10^{-4}$ and various n.*
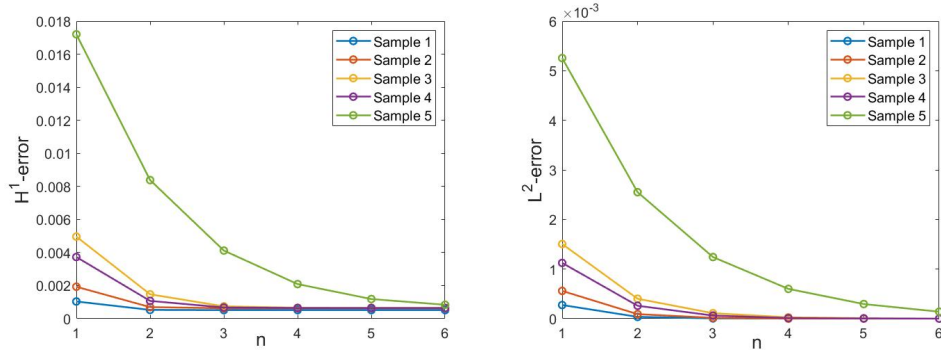

FIG. 6.10. *Approach #2: Errors $\mathcal{E}_{H^1}^n$ (left) and $\mathcal{E}_{L^2}^n$ (right) decays for first 5 samples with $\varepsilon = 2.0, a_0(x) = \mathbb{E}_{mc}[a(\cdot, x)] = 2.001, \rho = 0.5003, h = 0.01$ and various n.*

$f(\omega, x) = X(\omega)(51 - 100x)$ so that the exact solution is again given by

$$(6.17) \qquad u(\omega, x) = \frac{X(\omega)}{2(1 + \varepsilon X(\omega))}(x - x^2).$$

Its expected value is already given in previous subsection.

To apply the iterative method, we adopt Approach #2 proposed in Section 6.1 to (6.16). Namely, we consider the following parameter-dependent problem:

$$(6.18) \qquad -\frac{d}{dx}\Big(\big(1 + \varepsilon X(\omega_j)\big)\frac{du(\omega_j, x)}{dx}\Big) + 100\big(1 + \varepsilon X(\omega_j)\big)\frac{du(\omega_j, x)}{dx} = f(\omega_j, x),$$
$$u(\omega_j, 0) = 0, \qquad u(\omega_j, 1) = 0,$$

for $j = 1, \cdots, J$. To fit the setup of Section 6.1, we introduce the following coefficient functions:

$$
\begin{aligned}
a(\omega_j, x) &= 1 + \varepsilon X(\omega_j), & b(\omega_j, x) &= 100(1 + \varepsilon X(\omega_j)) \\
a_0(x) &= \mathbb{E}_{mc}[a(\cdot, x)], & b_0(x) &= \mathbb{E}_{mc}[b(\cdot, x)], \\
\eta_a(\omega_j, x) &= a(\omega_j, x) - a_0(x), & \eta_b(\omega_j, x) &= b(\omega_j, x) - b_0(x),
\end{aligned}
$$

which result in the following bilinear forms:

$$A(\omega_j; u, v) = \big(a(\omega_j, \cdot)u', v'\big) + \big(b(\omega_j, \cdot)u', v\big), \qquad F(\omega_j; v) = \big(f(\omega_j, \cdot), v\big),$$
$$A_0(u, v) = \big(a_0\, u', v'\big) + \big(b_0 u', v\big), \quad A_1(\omega_j; u, v) = \big(\eta_a(\omega_j, \cdot)u', v'\big) + \big(\eta_b(\omega_j, \cdot)u', v\big).$$

Then, the relative dominant number ρ in the convergence criteria of Section 3 can be estimated by

$$\rho \approx \frac{\max\limits_{1 \leq j \leq J} \max\limits_{x \in D}(|\eta_a(\omega_j, x)| + |\eta_b(\omega_j, x)|)}{\min\limits_{x \in D} a_0(x)}.$$

In this test, we use $J = 10^4$ number of samples for the Monte Carlo method, the mesh size $h = 0.01$ and the maximum number of iterations $n = 10$. We also select $\varepsilon = 0.2, 0.005$ in (6.18).
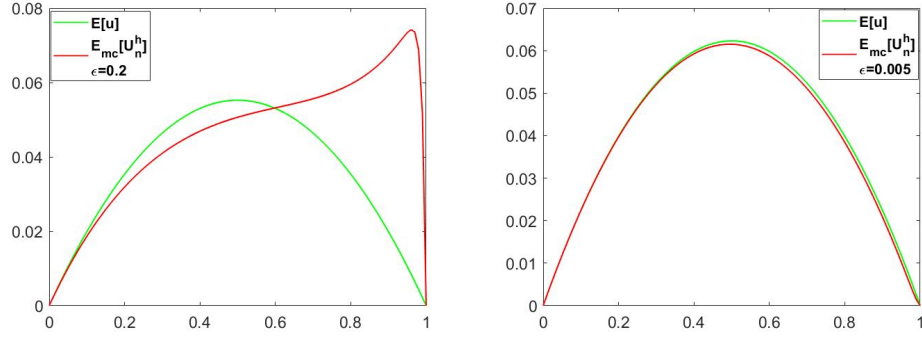

FIG. 6.11. *1-D random convection-diffusion problem: Plots of $\mathbb{E}[u]$ and $\mathbb{E}_{mc}[U_n^h]$ with $h = 0.01, n = 10$ and $\varepsilon = 0.2$ (left) and $\varepsilon = 0.005$(right).*

ε	ρ	$\mathcal{E}_{H^1}^n = \|\mathbb{E}[u] - \mathbb{E}_{mc}[U_n^h]\|_{H^1}$	$\mathcal{E}_{L^2}^n = \|\mathbb{E}[u] - \mathbb{E}_{mc}[U_n^h]\|_{L^2}$
0.2	9.1912	0.5328	2.2100×10^{-2}
0.005	0.2522	0.0117	9.6119×10^{-4}

TABLE 6.9
1-D random convection-diffusion problem: Errors $\mathcal{E}_{H^1}^n$ and $\mathcal{E}_{L^2}^n$ with $h = 0.01, n = 10$ and various ε.

Figure 6.11 shows the expected values $\mathbb{E}_{mc}[U_n^h]$ and $\mathbb{E}[u]$ of the computed and exact solutions and Table 6.9 displays the expected values of the H^1- and L^2-norm errors. We observe from Figure 6.11 and Table 6.9 that the proposed iterative method performs well for problem (6.16) when ε is sufficiently small, but the errors become larger as ε increases, which is expected. In addition, due to the contribution of the convection term in (6.16), the relative dominance parameter $\rho > 1$ when $\varepsilon = 0.2$, which explains the poor performance of the iterative method in this case.

6.4. 2-D randomized double-glazing problems. In this subsection, we consider the random perturbation of the double-glazing problem given in [18]. Specifically,

we consider

$$(6.19) \qquad -\delta \Delta u + \mathbf{b} \cdot \nabla u = f(\omega, x, y) \qquad \text{in } D = (0,1)^2$$

$$(6.20) \qquad u(\omega, \cdot) = 0 \qquad \text{on } \partial D,$$

where $\mathbf{b}(\omega, x, y) := (1 + \varepsilon X(\omega))\big(2y(1 - x^2), -2x(1 - y^2)\big)$ and $0 < \delta << |\mathbf{b}|$, $\varepsilon > 0$. X is a uniformly distributed random variable on $[0,1]$. It is easy to check that div $\mathbf{b} = 0$. We set the right hand-side force function

$$f(\omega; x, y) = \frac{\delta X(\omega)}{1 + \varepsilon X(\omega)} (y - y^2 + x - x^2) + X(\omega)y(y - y^2)(1 - x^2)(1 - 2x)$$
$$- X(\omega)x(x - x^2)(1 - y^2)(1 - 2y)$$

so that the exact solution is given by

$$u(\omega; x, y) = \frac{X(\omega)}{2(1 + \varepsilon X(\omega))}(x - x^2)(y - y^2).$$

Thus,

$$\mathbb{E}[u] = \frac{1}{2}\Big(\frac{1}{\varepsilon} - \frac{1}{\varepsilon^2}\ln(1 + \varepsilon)\Big)(x - x^2)(y - y^2).$$

We again adopt Approach #2 of Section 6.1 to solve (6.19) with $\delta = 0.1$ using the proposed iterative method with $J = 10^4$ number of Monte Carlo samples. Consequently, we need to solve the following parameter-dependent problem: for $1 \le j \le J$

$$(6.21) \qquad -\delta \Delta u(\omega_j, x, y) + \mathbf{b}(\omega_j, x, y) \cdot \nabla u(\omega_j, x, y) = f(\omega_j, x, y) \quad \text{in } D = (0,1)^2$$
$$u(\omega_j, \cdot) = 0 \qquad \text{on } \partial D.$$

To fit the setup of Section 6.1, we define

$$\mathbf{b}_0(x, y) := \mathbb{E}_{mc}[\mathbf{b}(\cdot, x, y)] = 2.001\big(2y(1 - x^2), -2x(1 - y^2)\big),$$
$$\boldsymbol{\eta}(\omega_j, x, y) := \mathbf{b}(\omega_j, x, y) - \mathbf{b}_0(x, y),$$

and the following bilinear forms and functional:

$$A(\omega_j; u, v) = \delta\big(\nabla u, \nabla v\big) + \big(\mathbf{b} \cdot \nabla u, v\big), \quad F(\omega_j; v) = \big(f, v\big),$$
$$A_0(u, v) = \delta\big(\nabla u, \nabla v\big) + \big(\mathbf{b}_0 \cdot \nabla u, v\big), \quad A_1(\omega_j; u, v) = \big(\boldsymbol{\eta} \cdot \nabla u, v\big).$$

h	$\mathcal{E}_{H^1}^n$	Order	$\mathcal{E}_{L^2}^n$	Order
0.2	5.4020×10^{-3}		3.4360×10^{-4}	
0.1	2.7339×10^{-3}	0.9825	8.2969×10^{-5}	2.0501
0.05	1.3709×10^{-3}	0.9958	1.7666×10^{-5}	2.2316
0.025	6.8635×10^{-4}	0.9806	3.8150×10^{-6}	2.2112

TABLE 6.10

2-D random convection-diffusion problem: Errors $\mathcal{E}_{H^1}^n$ and $\mathcal{E}_{L^2}^n$ with $\varepsilon = 2.0$, $n = 10$ and various h.

Table 6.10 and Figure 6.12 show respectively the convergence orders of the spatial discretization error and the iteration errors in both H^1- and L^2-norm. We observe that the proposed iterative method performs well for the random convection-dominated convection-diffusion problem (6.19). From Table 6.10 we see that the optimal convergence orders are achieved for the linear finite element method. In addition, Figure 6.12 shows that the H^1- and L^2-norm errors of the iterative solution rapidly decrease as the number of iterations increases.
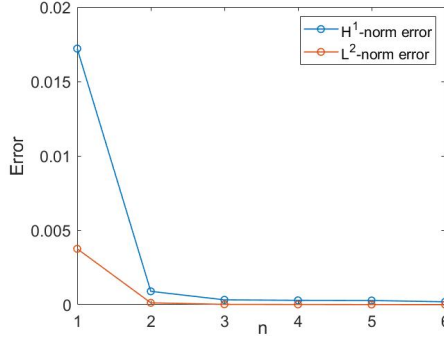

FIG. 6.12. *2-D convection-diffusion problem: Errors $\mathcal{E}_{H^1}^n$ and $\mathcal{E}_{L^2}^n$ with $h = 0.01, \varepsilon = 2.0$,* `tol` $= 10^{-4}$ *and various n.*

REFERENCES

[1] A. C. Antoulas, C. A. Beattie and S. Güğercin, *Interpolatory Methods for Model Reduction*, SIAM, Philadelphia, 2020.

[2] A. H. Baker, J. M. Dennis and E. R. Jessup, *On improving linear solver performance: a block variant of GMRES*, SIAM J. Sci. Comput., 27:1608–1626, 2006.

[3] S. C. Brenner and L.R Scott, *The Mathematical Theory of Finite Element Methods*, Third edition. Texts in Applied Mathematics, Vol. 15, Springer, New York, 2008.

[4] S. L. Brunton and J. N. Kutz, *Data-driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, London, 2019.

[5] T. A. Davis, *Direct Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2006.

[6] I. S. Duff, A.M. Erisman and J.K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, London, 2017.

[7] X. Feng, J. Lin and C. Lorton, *A multi-modes Monte Carlo finite element method for elliptic partial differential equations with random coefficients*, Int. J. Uncertain. Quanti. 6(5):429-443, 2016.

[8] X. Feng, J. Lin and C. Lorton, *An efficient numerical method for acoustic wave scattering in random media*, SIAM/ASA J. Uncertain. Quanti., 3:790-822, 2015.

[9] X. Feng, J. Lin and C. Lorton, *An efficient Monte Carlo interior penalty discontinuous Galerkin method for the time-harmonic Maxwell's Equations with random coefficients*, J. Sci. Comput., 80:1498-1528, 2019.

[10] A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10(2):345-363, 1973.

[11] M. Gunzburger, N. Jiang and M. Schneier, *An ensemble-proper orthogonal decomposition method for the nonstationary Navier–Stokes equations*, SIAM J. Numer. Anal., 55:286–304, 2017.

[12] M. Gunzburger, N. Jiang and Z. Wang, *An efficient algorithm for simulating ensembles of parameterized flow problems*, IMA J. Numer. Anal., 39:1180–1205, 2019.

[13] M. Gunzburger, N. Jiang and Z. Wang, *A second-order time-stepping scheme for simulating ensembles of parameterized flow problems*, Comput. Methods Appl. Math., 19:681–701, 2017.

[14] J. S. Hesthaven, G. Rozza and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer, New York, 2015.

[15] N. Jiang and W. Layton, *An algorithm for fast calculation of flow ensembles*, Int. J. Uncertain. Quanti., 4:273–301, 2014.

[16] L. Ju, W. Leng, Z. Wang and S. Yuan, *Numerical investigation of ensemble methods with block iterative solvers for evolution problems*, Discrete Contin. Dyn. Syst. Ser. B., 25(12):4905-4923, 2020.

[17] A. Quarteroni, A. Manzoni and F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, Springer, New York, 2015.

[18] D. Silvester, H. Elman and A. Ramage, *Incompressible Flow and Iterative Solver Software (IFISS), version 3.5*, 2016.

[19] W. C. Rheinboldt, *Numerical Analysis of Parametrized Nonlinear Equations*, Wiley, New York, 1986.

[20] Y. Luo and Z. Wang, *An ensemble algorithm for numerical solutions to deterministic and random parabolic PDEs*, SIAM J. Numer. Anal., 56(2):859-876, 2018.

[21] Y. Luo and Z. Wang, *A multilevel Monte Carlo ensemble scheme for solving random parabolic PDEs*, SIAM J. Sci. Comput., 41:A622–A642, 2019.

[22] Q. Du, V. Faber and M. Gunzburger, *Centroidal Voronoi tessellations: Applications and algorithms*, SIAM review, 41(4):637-676, 1999.