Efficient adaptive algorithms for elliptic PDEs with random data *

Alex Bespalov[†] Leonardo Rocchi[†]

Abstract

We present a novel adaptive algorithm implementing the stochastic Galerkin finite element method for numerical solution of elliptic PDE problems with correlated random data. The algorithm employs a hierarchical a posteriori error estimation strategy which also provides effective estimates of the error reduction for enhanced approximations. These error reduction indicators are used in the algorithm to perform a balanced adaptive refinement of spatial and parametric components of Galerkin approximations. The results of numerical tests demonstrating the efficiency of the algorithm for three representative PDEs with random coefficients are reported. The software used for numerical experiments is available online.

Key words: stochastic Galerkin methods, stochastic finite elements, PDEs with random data, adaptive methods, a posteriori error estimation, singularities, parametric PDEs

AMS Subject Classification: 35R60, 65C20, 65N30, 65N12, 65N50

1 Introduction

The development of efficient numerical algorithms for PDE problems with correlated random data is important for reliable uncertainty quantification. One of the methods that is commonly used in this context is the stochastic Galerkin finite element method (sGFEM). In this method, a parametric reformulation of the given PDE with random data is discretized, and approximations are sought in tensor product spaces $X \otimes \mathcal{P}$, where X is a finite element space associated with a physical domain and $\mathcal{P}$ is a set of multivariate polynomials over a finite-dimensional manifold in the parameter domain. If a large number of random variables is used to represent the input data and highly refined spatial grids are used for finite element approximations on the physical domain, then computing the sGFEM solution becomes prohibitively expensive, due to huge dimension of the space $X \otimes \mathcal{P}$. One way to avoid this is to use an adaptive approach, in which spatial (X-) and stochastic ($\mathcal{P}$-) components of approximations are judiciously chosen and incrementally refined in the course of numerical computation.

Although adaptive stochastic Galerkin finite element methods are still in the beginning of their development, there have been several very recent works that addressed the design and analysis aspects of these methods (see [10, 5, 11, 7, 12, 13]). In particular,

[*] This work was supported by the EPSRC under grant EP/P013791/1.

[†] School of Mathematics, University of Birmingham, Edgbaston, Birmingham B15 2TT (a.bespalov@bham.ac.uk, lxr507@bham.ac.uk).

the adaptive algorithms developed in [10, 11] are driven by spatial and stochastic error indicators derived from explicit residual-based a posteriori error estimators, whereas local equilibration error estimators are employed in [12].

The main focus of this paper is on designing efficient adaptive sGFEM algorithms that ensure a balanced refinement of spatial and stochastic components. Similarly to [10, 11, 12], our adaptive algorithm exploits the energy orthogonality inherent to Galerkin projection methods and employs Dörfler marking [9] for both spatial finite elements and stochastic basis functions. However, in contrast to these works, we build upon our results in [5, 7] and use a posteriori estimates of error reduction to steer adaptive refinement. Another novelty of our adaptive algorithm is how the balance between spatial and stochastic approximations is ensured. It is common to perform either spatial or stochastic refinement at each iteration of the algorithm. Traditionally, the choice between the two refinements is based on the dominant error estimator contributing to the *total* error estimate, cf. [10, 11, 7, 12, 13]; we employ this strategy in Version 1 of our adaptive algorithm. An alternative strategy is implemented in Version 2 of the algorithm: here, the refinement type is chosen by comparing the error reduction estimates for *marked* finite elements and for *marked* stochastic basis functions.

In this paper, we use representative parametric PDE problems posed over square, L-shaped, and slit domains to perform a systematic numerical study of the two versions of the adaptive algorithm. In particular, we compare the performance and convergence properties of both versions and reveal how the choice of Dörfler marking parameters (for both spatial refinement and polynomial enrichment) affects the performance of the algorithm for spatially regular and spatially singular problems.

An outline of the paper is as follows. Sections 2 introduces the model problem that can be seen as a parametric reformulation of a representative elliptic PDE with random data. In Sections 3 and 4, we describe the stochastic Galerkin FEM for numerical solution of the model problem, recall the construction of hierarchical a posteriori error estimators and the associated error reduction indicators, and discuss computational aspects of our error estimation strategy. In Section 5, two versions of the new adaptive algorithm are presented and described in detail. The results of numerical experiments are reported and analyzed in Section 6.

2 Parametric model problem

Let $D \subset \mathbb{R}^2$ be a bounded (spatial) domain with a Lipschitz polygonal boundary ∂D, and let $\Gamma := \prod_{m=1}^{\infty} \Gamma_m$ be the parameter domain, with Γ_m being bounded intervals in $\mathbb{R}$. Let $H_0^1(D)$ be the usual Sobolev space of functions in $H^1(D)$ vanishing at the boundary ∂D in the sense of traces, and let $\langle \cdot, \cdot \rangle$ denote the duality pairing between $H_0^1(D)$ and its dual space $H^{-1}(D)$. We consider the homogeneous Dirichlet problem for the parametric steady-state diffusion equation

$$\begin{aligned} -\nabla \cdot (a(\mathbf{x},\mathbf{y}) \nabla u(\mathbf{x},\mathbf{y})) &= f(\mathbf{x}), \qquad && \mathbf{x} \in D,\ \mathbf{y} \in \Gamma, \\ u(\mathbf{x},\mathbf{y}) &= 0, && \mathbf{x} \in \partial D,\ \mathbf{y} \in \Gamma, \end{aligned} \tag{2.1}$$

where $f \in H^{-1}(D)$, ∇ denotes differentiation with respect to spatial variables only, and the parametric diffusion coefficient is represented as

$$a(\mathbf{x},\mathbf{y}) = a_0(\mathbf{x}) + \sum_{m=1}^{\infty} y_m a_m(\mathbf{x}), \quad \mathbf{x} \in D,\ \mathbf{y} = (y_1, y_2, \ldots) \in \Gamma, \tag{2.2}$$

for a family of spatially varying functions $a_m(\mathbf{x})$, $m \in \mathbb{N}_0$, and with the series converging uniformly in $L^\infty(D)$.

As an example, decomposition (2.2) may come from a Karhunen-Loève expansion of a random field with given covariance function (see, e.g., [17, 2, 22, 19]). In this case, the parameter-free term $a_0(\mathbf{x})$ in (2.2) represents the mean of the random field, the coefficients $a_m(\mathbf{x})$, $m \in \mathbb{N}$, are orthogonal in $L^2(D)$, and the parameters y_m, $m \in \mathbb{N}$, are the images of i.i.d. mean-zero random variables with unit variance. One can ensure that these bounded random variables take values in $[-1, 1]$ by rescaling the functions $a_m(x)$ (see [22, Lemma 2.20]). Therefore, in what follows, we will assume that $\Gamma_m := [-1, 1]$ for all $m \in \mathbb{N}$.

Convergence of the series in (2.2) and positivity of $a(\mathbf{x}, \mathbf{y})$ for each $\mathbf{x} \in D$ and $\mathbf{y} \in \Gamma$ are ensured by making the following assumptions:

(i) we suppose that $a_0(\mathbf{x}) \in L^\infty(D)$ is uniformly bounded away from zero, that is there exist two constants $\alpha_0^{\min}$, $\alpha_0^{\max}$ such that

$$0 < \alpha_0^{\min} \le a_0(\mathbf{x}) \le \alpha_0^{\max} \quad \text{a.e. in } D; \tag{2.3}$$

(ii) we assume that $a_m(\mathbf{x}) \in L^\infty(D)$, $m \in \mathbb{N}$, are such that

$$\tau := \frac{1}{\alpha_0^{\min}} \sum_{m=1}^{\infty} \|a_m\|_{L^\infty(D)} < 1. \tag{2.4}$$

Now, for all $\mathbf{y} \in \Gamma$, we can define the linear operator $A(\mathbf{y}) \in \mathcal{L}(H_0^1(D), H^{-1}(D))$ as follows:

$$\langle A(\mathbf{y})v, w\rangle := \int_D a(\mathbf{x}, \mathbf{y}) \nabla v(\mathbf{x}) \cdot \nabla w(\mathbf{x})\, d\mathbf{x} \quad \forall\, v, w \in H_0^1(D). \tag{2.5}$$

Because of expansion (2.2), the operator $A(\mathbf{y})$ admits the decomposition

$$A(\mathbf{y}) = A_0 + \sum_{m=1}^{\infty} y_m A_m, \quad \mathbf{y} \in \Gamma, \tag{2.6}$$

where $A_m \in \mathcal{L}(H_0^1(D), H^{-1}(D))$, $m \in \mathbb{N}_0$, are defined by

$$\langle A_m v, w\rangle := \int_D a_m(\mathbf{x}) \nabla v(\mathbf{x}) \cdot \nabla w(\mathbf{x})\, d\mathbf{x} \quad \forall\, v, w \in H_0^1(D). \tag{2.7}$$

It easy to see that the assumption on $a_0(\mathbf{x})$ (see (2.3)) implies that $\langle A_0 v, w\rangle$ is a symmetric, continuous and coercive bilinear form, i.e.,

$$\begin{aligned} |\langle A_0 v, w\rangle| &\le \alpha_0^{\max} \|v\|_{H_0^1(D)} \|w\|_{H_0^1(D)} && \forall\, v, w \in H_0^1(D), \\ \langle A_0 v, v\rangle &\ge \alpha_0^{\min} \|v\|_{H_0^1(D)}^2 && \forall\, v \in H_0^1(D), \end{aligned} \tag{2.8}$$

whereas the assumption on coefficients $a_m(\mathbf{x})$ (see (2.4)) ensures convergence of the series in (2.6) in $\mathcal{L}(H_0^1(D), H^{-1}(D))$ uniformly in $\mathbf{y}$; see [22, Lemma 2.21]. Furthermore, (2.3) and (2.4) together imply the boundedness of both $A(\mathbf{y})$ and $A(\mathbf{y})^{-1}$ for all $\mathbf{y} \in \Gamma$, i.e.,

$$\sup_{\mathbf{y}\in\Gamma} \|A(\mathbf{y})\|_{\mathcal{L}(H_0^1, H^{-1})} \le \alpha_{\max} \quad \text{and} \quad \sup_{\mathbf{y}\in\Gamma} \|A(\mathbf{y})^{-1}\|_{\mathcal{L}(H^{-1}, H_0^1)} \le \alpha_{\min}^{-1}, \tag{2.9}$$

where $\alpha_{\max} := \alpha_0^{\max}(1+\tau)$ and $\alpha_{\min} := \alpha_0^{\min}(1-\tau)$; see [22, Proposition 2.22].

Let us now write the weak formulation of problem (2.1). To this end, we introduce a measure $\pi = \pi(\mathbf{y})$ on $(\Gamma, \mathcal{B}(\Gamma))$, where $\mathcal{B}(\Gamma)$ is the Borel σ-algebra on Γ. We assume that π is a product measure given by

$$\pi(\mathbf{y}) = \prod_{m=1}^{\infty} \pi_m(y_m), \quad \mathbf{y} \in \Gamma, \tag{2.10}$$

where each π_m is a symmetric probability measure on $(\Gamma_m, \mathcal{B}(\Gamma_m))$, with $\mathcal{B}(\Gamma_m)$ representing the Borel σ-algebra on Γ_m. Then $L^2_\pi(\Gamma)$ represents the Lebesgue space of functions $v \colon \Gamma \to \mathbb{R}$ that are square integrable on Γ with respect to the measure π, and $\langle \cdot, \cdot \rangle_\pi$ denotes the associated inner product. We will denote by $V := L^2_\pi(\Gamma; H^1_0(D))$ the Bochner space of strongly measurable functions $v \colon D \times \Gamma \to \mathbb{R}$ such that

$$\|v\|_V := \left(\int_\Gamma \|v(\cdot, \mathbf{y})\|^2_{H^1_0(D)} d\pi(\mathbf{y}) \right)^{1/2} < +\infty.$$

Then the weak formulation of (2.1) reads as follows: find $u \in V$ such that

$$B(u, v) = F(v) \quad \forall\, v \in V, \tag{2.11}$$

with the symmetric bilinear form and the linear functional given by

$$B(u, v) := \int_\Gamma \langle A(\mathbf{y}) u(\mathbf{y}), v(\mathbf{y}) \rangle \, d\pi(\mathbf{y}) \quad \text{and} \quad F(v) := \int_\Gamma \langle f, v(\mathbf{y}) \rangle d\pi(\mathbf{y}). \tag{2.12}$$

By using decomposition (2.6), we can rewrite the bilinear form in (2.12) as

$$B(u, v) := B_0(u, v) + \sum_{m=1}^{\infty} B_m(u, v) \quad \forall\, u, v \in V, \tag{2.13}$$

with the component bilinear forms $B_m(\cdot, \cdot)$, $m \in \mathbb{N}_0$, defined as

$$B_0(u, v) := \int_\Gamma \langle A_0 u(\mathbf{y}), v(\mathbf{y}) \rangle \, d\pi(\mathbf{y}), \tag{2.14}$$

$$B_m(u, v) := \int_\Gamma \langle A_m(\mathbf{y}), v(\mathbf{y}) \rangle \, y_m \, d\pi(\mathbf{y}) \quad \forall\, m \in \mathbb{N}. \tag{2.15}$$

It is evident that inequalities (2.9) imply that $B(\cdot, \cdot)$ is continuous and coercive with $\alpha_{\max}$ and $\alpha_{\min}$ being the continuity and coercivity constants, respectively. Furthermore, $f \in L^2_\pi(\Gamma; H^{-1}(D))$. Therefore, the existence of the unique solution $u \in V$ satisfying (2.11) is guaranteed by the Lax-Milgram lemma.

We observe that $B(\cdot, \cdot)$ defines an inner product in V which induces the norm $\|v\|_B := B(v, v)^{1/2}$ that is equivalent to $\|v\|_V$. On the other hand, inequalities (2.8) imply that $B_0(\cdot, \cdot)$ given by (2.14) also defines an inner product in V inducing the norm $\|v\|_{B_0} := B_0(v, v)^{1/2}$ which is equivalent to $\|v\|_V$. Therefore, the norms induced by B and B_0 are equivalent, i.e., the following two-sided inequality holds

$$\lambda\, B(v, v) \le B_0(v, v) \le \Lambda\, B(v, v) \quad \forall\, v \in V, \tag{2.16}$$

with the constants $\lambda := \alpha_0^{\min}/\alpha_{\max}$ and $\Lambda := \alpha_0^{\max}/\alpha_{\min}$.

3 Stochastic Galerkin discretizations

We introduce now the stochastic Galerkin Finite Element Method for discretization of the weak formulation (2.11). For any finite-dimensional subspace $V_N \subset V$, problem (2.11) can be discretized by using Galerkin projection onto V_N. This defines a unique function $u_N \in V_N$ satisfying

$$B(u_N, v) = F(v) \quad \forall\, v \in V_N.$$

The starting point in constructing the finite-dimensional subspace $V_N \subset V$ is to notice that the Bochner space $V = L^2_\pi(\Gamma; H^1_0(D))$ is isometrically isomorphic to the tensor product Hilbert space $H^1_0(D) \otimes L^2_\pi(\Gamma)$ (see, e.g., [22, Theorem B.17, Remark C.24]). Therefore, V_N can be defined by mimicking this tensor product structure. More specifically, we construct two approximation spaces $X \subset H^1_0(D)$ and $\mathcal{P}_\mathfrak{P} \subset L^2_\pi(\Gamma)$ independently of each other and then define $V_N := X \otimes \mathcal{P}_\mathfrak{P} \subset H^1_0(D) \otimes L^2_\pi(\Gamma) \cong V$.

For the finite-dimensional subspace of $H^1_0(D)$, we will use the finite element space $X = X(\mathcal{T})$ of continuous piecewise linear functions on a shape-regular and conforming triangulation $\mathcal{T}$ of D.

Let us now introduce the finite-dimensional (polynomial) subspaces of $L^2_\pi(\Gamma)$. For each $m \in \mathbb{N}$, let $\{p^m_n\}_{n\in\mathbb{N}_0}$ denote the set of univariate polynomials on $\Gamma_m = [-1, 1]$ that are orthonormal with respect the inner product $\langle\cdot,\cdot\rangle_{\pi_m}$ in $L^2_{\pi_m}(\Gamma_m)$. For any polynomial p^m_n, the subscript n indicates the polynomial degree and we denote by c^m_n the leading coefficient of p^m_n. The set $\{p^m_n\}_{n\in\mathbb{N}_0}$ is an orthonormal basis of the space $L^2_{\pi_m}(\Gamma_m)$. Furthermore, as a consequence of the symmetry of π_m, these polynomials satisfy the following three-term recurrence (e.g., see [16]):

$$p^m_0 \equiv 1; \quad \beta^m_{n+1}\, p^m_{n+1}(y_m) = y_m\, p^m_n(y_m) - \beta^m_n\, p^m_{n-1}(y_m), \quad y_m \in \Gamma_m,\, n \in \mathbb{N}, \tag{3.1}$$

where $\beta^m_n = c^m_{n-1}/c^m_n$ for $n \in \mathbb{N}$. For example, if y_m are the images of uniformly distributed mean-zero random variables, that is π_m satisfies $d\pi_m(y_m) = \frac{1}{2}dy_m$, then $\{p^m_n\}_{n\in\mathbb{N}_0}$ is the set of scaled Legendre polynomials which are orthonormal with respect to $\langle\cdot,\cdot\rangle_{\pi_m}$ and $\beta^m_n = n(4n^2-1)^{-1/2}$.

In order to construct an orthonormal basis for the space $L^2_\pi(\Gamma)$, we introduce the following set of finitely supported sequences:

$$\mathfrak{I} := \left\{\nu = (\nu_1, \nu_2, \dots) \in \mathbb{N}_0^\mathbb{N};\ \#\mathrm{supp}(\nu) < \infty\right\}, \tag{3.2}$$

where $\mathrm{supp}(\nu) := \{m \in \mathbb{N};\ \nu_m \neq 0\}$ and $\#$ denotes the cardinality of a set. The set $\mathfrak{I}$ and any of its subsets will be called *index sets*, and their elements $\nu \in \mathfrak{I}$ will be called *indices*. Then we consider the following tensor product polynomials:

$$P_\nu(\mathbf{y}) := \prod_{m=1}^\infty p^m_{\nu_m}(y_m) = \prod_{m\in\mathrm{supp}(\nu)} p^m_{\nu_m}(y_m) \quad \forall\, \nu \in \mathfrak{I} \tag{3.3}$$

(here, the last equality holds, because $p^m_0 \equiv 1$ for any $m \in \mathbb{N}$). The countable set of these multivariate polynomials indexed by $\nu \in \mathfrak{I}$ forms an orthonormal basis of $L^2_\pi(\Gamma)$ (see, e.g., [19, Theorem 9.55]).

Given a finite index set $\mathfrak{P} \subset \mathfrak{I}$, the space of tensor product polynomials

$$\mathcal{P}_\mathfrak{P} := \mathrm{span}\{P_\nu;\ \nu \in \mathfrak{P}\} \tag{3.4}$$

defines a finite-dimensional subspace of $L^2_\pi(\Gamma)$, and its dimension is $\dim(\mathcal{P}_\mathfrak{P}) = \#\mathfrak{P}$. Note that each $P_\nu \in \mathcal{P}_\mathfrak{P}$ is a polynomial in a finite number of 'active' parameters y_m for which the corresponding ν_m is nonzero. Moreover, the nonzero values ν_m determine the polynomial degrees in these 'active' parameters.

With both spaces $X \subset H_0^1(D)$ and $\mathcal{P}_{\mathfrak{P}} \subset L^2_\pi(\Gamma)$ at hand, we can now define the finite-dimensional subspace $V_N = V_{X\mathfrak{P}} := X \otimes \mathcal{P}_{\mathfrak{P}}$ and rewrite the discrete formulation of (2.11) in the following way: find $u_{X\mathfrak{P}} \in V_{X\mathfrak{P}}$ such that

$$B(u_{X\mathfrak{P}}, v) = F(v) \quad \forall\, v \in V_{X\mathfrak{P}}. \tag{3.5}$$

Hereafter we implicitly assume that $\mathfrak{P}$ always contains the zero-index $\mathbf{0} := (0, 0, \dots)$.

The approximation provided by $u_{X\mathfrak{P}}$ can be improved by computing the Galerkin solution $u^*_{X\mathfrak{P}}$ in the enhanced subspace $V^*_{X\mathfrak{P}} \supset V_{X\mathfrak{P}}$. The enhanced subspace can be obtained by enriching the finite element space $X \subset H_0^1(D)$ and/or the polynomial space $\mathcal{P}_{\mathfrak{P}} \subset L^2_\pi(\Gamma)$. Let $X^* \subset H_0^1(D)$ denote the enriched finite element space such that $X^* \supset X$. The space X^*, for example, can be obtained from X by adding new piecewise linear basis functions corresponding to the nodes introduced by a uniform refinement of $\mathcal{T}$ (see, e.g., [1, Figure 5.2] for examples of such basis functions). Alternatively, the space X^* can be constructed by augmenting X with higher-order basis functions on the same triangulation $\mathcal{T}$. In both cases, the enhanced X^* can be decomposed as

$$X^* = X \oplus Y, \tag{3.6}$$

where Y is called the *detail space* and it satisfies $Y \subset H_0^1(D)$ and $X \cap Y = \{0\}$.

Since $\langle A_0 \cdot, \cdot \rangle$ defines an inner product in $H_0^1(D)$, it is well-known that there exists a positive constant $\gamma \in [0, 1)$ depending only on X and Y such that the strengthened Cauchy-Schwarz inequality holds (e.g., see [1, 14])

$$|\langle A_0 u_X, v_Y \rangle| \le \gamma\, \langle A_0 u_X, u_X \rangle^{1/2} \langle A_0 v_Y, v_Y \rangle^{1/2} \quad \forall\, u_X \in X,\ \forall\, v_Y \in Y. \tag{3.7}$$

The polynomial space $\mathcal{P}_{\mathfrak{P}}$ can be enriched by 'activating' new parameters y_m and/or by including higher-order polynomials in the 'active' parameters in $\mathfrak{P}$. This is done by constructing an enriched index set $\mathfrak{P}^* = \mathfrak{P} \cup \mathfrak{Q}$ with $\mathfrak{Q} \subset \mathfrak{I}$ satisfying $\mathfrak{P} \cap \mathfrak{Q} = \emptyset$. We will call $\mathfrak{Q}$ the *detail index set*. Therefore, the enriched polynomial space $\mathcal{P}_{\mathfrak{P}^*}$ can be decomposed as

$$\mathcal{P}_{\mathfrak{P}^*} = \mathcal{P}_{\mathfrak{P}} \oplus \mathcal{P}_{\mathfrak{Q}}, \tag{3.8}$$

where $\mathcal{P}_{\mathfrak{Q}}$ represents the polynomial space associated with $\mathfrak{Q}$ such that $\mathcal{P}_{\mathfrak{P}} \cap \mathcal{P}_{\mathfrak{Q}} = \{0\}$. Note that decomposition (3.8) is orthogonal with respect to the inner product $\langle \cdot, \cdot \rangle_\pi$.

We use the finite-dimensional subspaces $X, Y \subset H_0^1(D)$ and the index sets $\mathfrak{P}, \mathfrak{Q} \subset \mathfrak{I}$ to define the following finite-dimensional tensor product spaces:

$$V_{Y\mathfrak{P}} := Y \otimes \mathcal{P}_{\mathfrak{P}} \quad \text{and} \quad V_{X\mathfrak{Q}} := X \otimes \mathcal{P}_{\mathfrak{Q}}. \tag{3.9}$$

Hence, we can define the enriched finite-dimensional subspace of V as

$$V^*_{X\mathfrak{P}} := V_{X\mathfrak{P}} \oplus V_{Y\mathfrak{P}} \oplus V_{X\mathfrak{Q}}. \tag{3.10}$$

Now, let $u^*_{X\mathfrak{P}} \in V^*_{X\mathfrak{P}}$ be the Galerkin projection onto the enriched space $V^*_{X\mathfrak{P}}$, so that

$$B(u^*_{X\mathfrak{P}}, v) = F(v) \quad \forall\, v \in V^*_{X\mathfrak{P}}.$$

As it is commonly done in the analysis of hierarchical a posteriori error estimators (see, e.g., [1, Chapter 5]), we will assume that the Galerkin solution $u^*_{X\mathfrak{P}} \in V^*_{X\mathfrak{P}}$ is indeed an improvement over $u_{X\mathfrak{P}} \in V_{X\mathfrak{P}}$, that is, there exists a constant $\beta \in [0, 1)$ such that

$$\|u - u^*_{X\mathfrak{P}}\|_B \le \beta \|u - u_{X\mathfrak{P}}\|_B. \tag{3.11}$$

Note that, since $V_{X\mathfrak{P}} \subset V^*_{X\mathfrak{P}}$, inequality (3.11) always holds with some $\beta \le 1$ due to the best approximation property of Galerkin projections. We also refer to [5, Remark 3.1] for the representation of β that holds due to the tensor product structure of approximation spaces $V_{X\mathfrak{P}}$ and $V^*_{X\mathfrak{P}}$.

4 Hierarchical a posteriori error estimators

In this section, we recall the construction of hierarchical a posteriori estimators for the discretization error $e := u - u_{X\mathfrak{P}} \in V$ (this error estimation strategy in the context of the sGFEM was first described in [5] and further developed in [7]). Similar to what is done in nonparametric a posteriori error analysis (see, e.g., [1, 25]), we project the weak formulation satisfied by the error onto the enriched space $V^*_{X\mathfrak{P}}$ given by (3.10) to obtain

$$B(e, v) = F(v) - B(u_{X\mathfrak{P}}, v) \quad \forall\, v \in V^*_{X\mathfrak{P}}. \tag{4.1}$$

Then, using the bilinear form $B_0(\cdot,\cdot)$ given by (2.14) instead of $B(\cdot,\cdot)$ on the left-hand side of (4.1) and exploiting the tensor product structure of $V^*_{X\mathfrak{P}}$, we consider the following two independent problems posed on the lower-dimensional subspaces $V_{Y\mathfrak{P}}$ and $V_{X\mathfrak{Q}}$ given by (3.9): find the *spatial* error estimator $e_{Y\mathfrak{P}} \in V_{Y\mathfrak{P}}$ and the *parametric* error estimator $e_{X\mathfrak{Q}} \in V_{X\mathfrak{Q}}$ such that

$$B_0(e_{Y\mathfrak{P}}, v) = F(v) - B(u_{X\mathfrak{P}}, v) \quad \forall\, v \in V_{Y\mathfrak{P}}, \tag{4.2}$$

$$B_0(e_{X\mathfrak{Q}}, v) = F(v) - B(u_{X\mathfrak{P}}, v) \quad \forall\, v \in V_{X\mathfrak{Q}}. \tag{4.3}$$

Combining together the two estimators, we define

$$\eta := (\, \|e_{Y\mathfrak{P}}\|^2_{B_0} + \|e_{X\mathfrak{Q}}\|^2_{B_0}\,)^{1/2}. \tag{4.4}$$

It is easy to see that $\eta = \|e_{Y\mathfrak{P}} + e_{X\mathfrak{Q}}\|_{B_0}$ due to the orthogonality of polynomial spaces $\mathcal{P}_{\mathfrak{P}}$ and $\mathcal{P}_{\mathfrak{Q}}$ with respect to the inner product $\langle\cdot,\cdot\rangle_\pi$. The following result shows that η is an efficient and reliable estimate for the energy norm of the discretization error $e := u - u_{X\mathfrak{P}}$.

Proposition 4.1 [7, Theorem 4.1] *Let $u \in V$ be the solution of* (2.11) *and let $u_{X\mathfrak{P}} \in V_{X\mathfrak{P}}$ be the Galerkin approximation satisfying* (3.5). *Suppose that the saturation assumption* (3.11) *holds. Then, the a posteriori error estimate η defined by* (4.4) *satisfies*

$$\sqrt{\lambda}\,\eta \le \|u - u_{X\mathfrak{P}}\|_B \le \frac{\sqrt{\Lambda}}{\sqrt{1-\beta^2}\,\sqrt{1-\gamma^2}}\,\eta, \tag{4.5}$$

where λ, Λ are the constants in (2.16), *$\gamma \in [0,1)$ is the constant in the strengthened Cauchy-Schwarz inequality* (3.7), *and $\beta \in [0,1)$ is the constant in* (3.11).

Remark 4.1 *Despite global reliability and efficiency of hierarchical error estimates (as in Proposition* 4.1*), the associated local error indicators (see* (4.10) *below) are, in general, not reliable in the sense that the effectivity indices (defined as the ratio of the error estimate to the true error in the energy norm, cf.* (6.3)*) may become less than unity (see, e.g.,* [3], [15, Section 1.5.2] *as well as the results of numerical experiments in Section* 6 *below).*

While it is evident from Proposition 4.1 that η given by (4.4) can be used to control the error in the Galerkin approximation at each iteration of the adaptive algorithm, it turns out that the component estimators $e_{Y\mathfrak{P}}$ and $e_{X\mathfrak{Q}}$ contributing to η can be used to guide the adaptive process. Indeed, it has been shown in [5] that the B_0-norm of $e_{Y\mathfrak{P}}$ (resp., the B_0-norm of $e_{X\mathfrak{Q}}$) provides an effective estimate of the error reduction that would be achieved if we were to enrich only the finite element space X (resp., the polynomial space $\mathcal{P}_{\mathfrak{P}}$) and to compute the corresponding enhanced approximation. Suppose, for instance, that the polynomial space $\mathcal{P}_{\mathfrak{P}}$ is enriched. Then, the corresponding enhanced approximation $u_{X\mathfrak{P}^*} \in V_{X\mathfrak{P}} \oplus V_{X\mathfrak{Q}} =: V_{X\mathfrak{P}^*}$ satisfies

$$B(u_{X\mathfrak{P}^*}, v) = F(v) \quad \forall\, v \in V_{X\mathfrak{P}^*}. \tag{4.6}$$

Since the bilinear form $B(\cdot,\cdot)$ is symmetric, the Pythagorean theorem and the Galerkin orthogonality yield the equality

$$\|e\|_B^2 = \|u - u_{X\mathfrak{P}^*}\|_B^2 + \|u_{X\mathfrak{P}^*} - u_{X\mathfrak{P}}\|_B^2.$$

This shows that the error reduction achieved by enriching only the space $\mathcal{P}_{\mathfrak{P}}$ is given by $\|u_{X\mathfrak{P}^*} - u_{X\mathfrak{P}}\|_B^2$. The same argument applies if we enrich only the finite element space X and compute the enhanced approximation $u_{X^*\mathfrak{P}} \in V_{X\mathfrak{P}} \oplus V_{Y\mathfrak{P}} =: V_{X^*\mathfrak{P}}$ satisfying

$$B(u_{X^*\mathfrak{P}}, v) = F(v) \quad \forall\, v \in V_{X^*\mathfrak{P}}. \tag{4.7}$$

In this case, the error reduction is given by the quantity $\|u_{X^*\mathfrak{P}} - u_{X\mathfrak{P}}\|_B^2$. The following result establishes the two-sided bounds for both error reductions.

Proposition 4.2 [5, Theorem 5.1] *Let $u_{X\mathfrak{P}}{\in}V_{X\mathfrak{P}}$ be the Galerkin approximation satisfying* (3.5)*, and let $u_{X^*\mathfrak{P}}{\in}V_{X^*\mathfrak{P}}$ and $u_{X\mathfrak{P}^*}{\in}V_{X\mathfrak{P}^*}$ be the enhanced approximations satisfying* (4.7) *and* (4.6)*, respectively. Then, the following estimates for the error reduction hold:*

$$\sqrt{\lambda}\|e_{Y\mathfrak{P}}\|_{B_0} \le \|u_{X^*\mathfrak{P}} - u_{X\mathfrak{P}}\|_B \le \frac{\sqrt{\Lambda}}{\sqrt{1-\gamma^2}}\|e_{Y\mathfrak{P}}\|_{B_0}, \tag{4.8}$$

$$\sqrt{\lambda}\|e_{X\mathfrak{Q}}\|_{B_0} \le \|u_{X\mathfrak{P}^*} - u_{X\mathfrak{P}}\|_B \le \sqrt{\Lambda}\|e_{X\mathfrak{Q}}\|_{B_0}, \tag{4.9}$$

where $e_{Y\mathfrak{P}} \in V_{Y\mathfrak{P}}$ and $e_{X\mathfrak{Q}} \in V_{X\mathfrak{Q}}$ are defined by (4.2) *and* (4.3)*, respectively, λ and Λ are the constants in* (2.16)*, and $\gamma \in [0, 1)$ is the constant in* (3.7)*.*

It is important to note that given the problem data and the computed Galerkin approximation $u_{X\mathfrak{P}}$, the two estimates $\|e_{Y\mathfrak{P}}\|_{B_0}$ and $\|e_{X\mathfrak{Q}}\|_{B_0}$ are computable for any finite-dimensional subspace $Y \subset H_0^1(D)$ and for any finite index set $\mathfrak{Q} \subset \mathfrak{I}$ such that $X \cap Y = \{0\}$ and $\mathfrak{P} \cap \mathfrak{Q} = \emptyset$. On the one hand, a combination of $\|e_{Y\mathfrak{P}}\|_{B_0}$ and $\|e_{X\mathfrak{Q}}\|_{B_0}$ provides a reliable and efficient estimate η for the energy error $\|u - u_{X\mathfrak{P}}\|_B$ (see Proposition 4.1). On the other hand, Proposition 4.2 shows that the error reductions $\|u_{X^*\mathfrak{P}} - u_{X\mathfrak{P}}\|_B$ and $\|u_{X\mathfrak{P}^*} - u_{X\mathfrak{P}}\|_B$ can be estimated by $\|e_{Y\mathfrak{P}}\|_{B_0}$ and $\|e_{X\mathfrak{Q}}\|_{B_0}$, respectively. We emphasize that our choice of the detail subspace Y and the detail index set $\mathfrak{Q}$ may depend on whether we want to estimate the error in the Galerkin approximation $u_{X\mathfrak{P}}$ or estimate the error reduction achieved by enhancing this approximation. In particular, we can employ a large detail space $Y \subset H_0^1(D)$ (e.g., based on a uniform refinement of the current triangulation) and a large detail index set $\mathfrak{Q} \subset \mathfrak{I} \setminus \mathfrak{P}$ in order to obtain an accurate estimate of the error $\|u - u_{X\mathfrak{P}}\|_B$. Computational aspects of this procedure are discussed next.

4.1 Computational aspects of the error estimation

Let $u_{X\mathfrak{P}} \in V_{X\mathfrak{P}} = X \otimes \mathcal{P}_{\mathfrak{P}}$ be the computed Galerkin solution satisfying (3.5), where $X = X(\mathcal{T})$ is the space of continuous piecewise linear functions associated with a triangulation $\mathcal{T}$ and $\mathcal{P}_{\mathfrak{P}}$ is the polynomial space on Γ (see (3.4)) associated with an index set $\mathfrak{P}$. We will denote by N the overall number of degrees of freedom, which is given by

$$N = \dim\left(X(\mathcal{T}) \otimes \mathcal{P}_{\mathfrak{P}}\right) = \dim(X(\mathcal{T})) \cdot \dim(\mathcal{P}_{\mathfrak{P}}) = \#\mathcal{N} \cdot \#\mathfrak{P},$$

where $\mathcal{N}$ denotes the set of interior vertices of the triangulation $\mathcal{T}$.

We will now describe how the estimate η (see (4.4)) for the energy error $\|u - u_{X\mathfrak{P}}\|_B$ is actually computed. The spatial estimator $e_{Y\mathfrak{P}} \in V_{Y\mathfrak{P}} = Y \otimes \mathcal{P}_{\mathfrak{P}}$ contributing to η satisfies the discrete formulation (4.2). In our computations, we choose the finite element detail space Y as the span of linear Lagrange basis functions defined at the edge midpoints of $\mathcal{T}$ and corresponding to the uniform refinement of $\mathcal{T}$ obtained using the criss-cross subdivision (see [1, Figure 5.2]). In order to solve (4.2) we use a standard element residual technique (e.g., see [1]). Specifically, on each element $K \in \mathcal{T}$ we compute the local (spatial) error estimator by solving the local residual problem associated with (4.2): find $e_{Y\mathfrak{P}}|_K \in V_{Y\mathfrak{P}}|_K$ satisfying

$$\begin{aligned} B_{0,K}(e_{Y\mathfrak{P}}|_K, v) = F_K(v) &+ \int_\Gamma \int_K \nabla \cdot \left(a(\mathbf{x},\mathbf{y}) \nabla u_{X\mathfrak{P}}(\mathbf{x},\mathbf{y})\right)\, v(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\pi(\mathbf{y}) \\ &- \frac{1}{2} \int_\Gamma \int_{\partial K \setminus \partial D} a(s,\mathbf{y}) \left[\!\!\left[\frac{\partial u_{X\mathfrak{P}}}{\partial n} \right]\!\!\right] v(s,\mathbf{y})\, ds\, d\pi(\mathbf{y}) \end{aligned} \tag{4.10}$$

for any $v \in V_{Y\mathfrak{P}}|_K$. Here $V_{Y\mathfrak{P}}|_K := Y|_K \otimes \mathcal{P}_{\mathfrak{P}}$ with $Y|_K$ being the restriction of Y to the element $K \in \mathcal{T}$, $B_{0,K}(\cdot,\cdot)$ and $F_K(\cdot)$ denote the elementwise bilinear form and the linear functional, respectively, and $\left[\!\left[\frac{\partial u_{X\mathfrak{P}}}{\partial n}\right]\!\right]$ represents the flux jump in the Galerkin solution $u_{X\mathfrak{P}}$ across interelement edges. There are two important features of this error estimation technique (see [5, Section 6.2] for details): (i) the linear algebra associated with solving (4.10) is very simple; (ii) the computation of error estimators can be vectorized or parallelized over finite elements in $\mathcal{T}$.

Turning now to the parametric estimator $e_{X\mathfrak{Q}} \in V_{X\mathfrak{Q}} = X \otimes \mathcal{P}_{\mathfrak{Q}}$ defined by (4.3), we exploit the theoretical results obtained in [7, Section 4.2] in order to choose an appropriate detail index set $\mathfrak{Q}$ and to decompose $e_{X\mathfrak{Q}}$ into contributions from the estimators associated with individual indices $\mu \in \mathfrak{Q}$.

First, for a given index set $\mathfrak{P}$, let us consider the infinite index set

$$\mathfrak{Q}_\infty := \left\{\mu \in \mathfrak{I} \setminus \mathfrak{P};\ \mu = \nu \pm \varepsilon^{(m)}\ \forall\, \nu \in \mathfrak{P},\ \forall\, m \in \mathbb{N}\right\},$$

where $\varepsilon^{(m)} := (\varepsilon_1^{(m)}, \varepsilon_2^{(m)}, \dots)$ denotes the Kronecker delta sequence such that $\varepsilon_j^{(m)} = \delta_{mj}$, for all $j \in \mathbb{N}$. It turns out that for any finite index set $\mathfrak{Q} \subset \mathfrak{I} \setminus (\mathfrak{P} \cup \mathfrak{Q}_\infty)$ the error estimator $e_{X\mathfrak{Q}}$ is identically zero (see Lemma 4.3 and Corollary 4.1 in [7]). This motivates our choice of the (finite) detail index set $\mathfrak{Q} \subset \mathfrak{Q}_\infty$ to be used for computing $e_{X\mathfrak{Q}}$:

$$\mathfrak{Q} := \left\{\mu \in \mathfrak{I} \setminus \mathfrak{P};\ \mu = \nu \pm \varepsilon^{(m)}\ \forall\, \nu \in \mathfrak{P},\ \forall\, m = 1, \dots, M_{\mathfrak{P}} + 1\right\}, \tag{4.11}$$

where the counter parameter $M_{\mathfrak{P}}$ is defined as follows:

$$M_{\mathfrak{P}} := \begin{cases} 0 & \text{if } \mathfrak{P} = \{\mathbf{0}\}, \\ \max\left\{\max(\mathrm{supp}(\nu));\ \nu \in \mathfrak{P} \setminus \{\mathbf{0}\}\right\} & \text{otherwise.} \end{cases}$$

Then, for each index $\mu \in \mathfrak{Q}$, we compute the estimator $e_{X\mathfrak{Q}}^{(\mu)} \in X \otimes \mathcal{P}_\mu$ by solving the linear system associated with the following discrete formulation:

$$B_0(e_{X\mathfrak{Q}}^{(\mu)}, v) = F(v) - B(u_{X\mathfrak{P}}, v) \quad \forall\, v \in X \otimes \mathcal{P}_\mu. \tag{4.12}$$

Two important observations are due here: (i) the coefficient matrix of the linear system associated with (4.12) is the same for all $\mu \in \mathfrak{Q}$; thus, the estimators $e_{X\mathfrak{Q}}^{(\mu)}$ can be computed efficiently by factorizing this matrix and performing $\#\mathfrak{Q}$ independent forward and backward substitutions; (ii) by Proposition 4.2, the norm $\|e_{X\mathfrak{Q}}^{(\mu)}\|_{B_0}$ provides an estimate of the error reduction that would be achieved by computing the enhanced Galerkin approximation $u_{X\mathfrak{P}^*} \in X \otimes (\mathcal{P}_\mathfrak{P} \oplus \mathcal{P}_\mu)$ (i.e., by adding only one new basis function P_μ to the polynomial space on Γ).

Finally, by using [7, Lemma 4.2], the overall parametric error estimator $e_{X\mathfrak{Q}}$ and its norm $\|e_{X\mathfrak{Q}}\|_{B_0}$ are computed from the contributions associated with individual indices in $\mathfrak{Q}$ as follows:

$$e_{X\mathfrak{Q}} = \sum_{\mu \in \mathfrak{Q}} e_{X\mathfrak{Q}}^{(\mu)}, \qquad \|e_{X\mathfrak{Q}}\|_{B_0} = \left(\sum_{\mu \in \mathfrak{Q}} \|e_{X\mathfrak{Q}}^{(\mu)}\|_{B_0}^2 \right)^{1/2}.$$

Once all elementwise estimators $e_{Y\mathfrak{P}}|_K$ $(K \in \mathcal{T})$ and all contributing parametric estimators $e_{X\mathfrak{Q}}^{(\mu)}$ $(\mu \in \mathfrak{Q})$ are computed, the total error estimate η is calculated via

$$\eta = \left(\sum_{K \in \mathcal{T}} \left\| e_{Y\mathfrak{P}}|_K \right\|_{B_{0,K}}^2 + \sum_{\mu \in \mathfrak{Q}} \left\| e_{X\mathfrak{Q}}^{(\mu)} \right\|_{B_0}^2 \right)^{1/2}, \tag{4.13}$$

where $\|\cdot\|_{B_{0,K}}^2 = B_{0,K}(\cdot,\cdot)$.

5 Adaptive algorithm for the sGFEM

In this section, we present an adaptive sGFEM algorithm for numerical solution of the parametric diffusion problem (2.1). The algorithm can be extended in an obvious way to other parametric PDE problems with affine dependence on (random) parameters. The algorithm follows the standard adaptive iteration: starting with a coarse triangulation $\mathcal{T}_0$ and an initial index set $\mathfrak{P}_0$ it generates a sequence of nested finite element spaces $\{X(\mathcal{T}_k)\}_{k \ge 0}$ $(X(\mathcal{T}_k) \subseteq X(\mathcal{T}_{k+1}) \subset H_0^1(D))$, a sequence of nested index sets $\{\mathfrak{P}_k\}_{k \ge 0}$ $(\mathfrak{P}_k \subseteq \mathfrak{P}_{k+1} \subset \mathfrak{I})$, and a sequence of refined sGFEM approximations $\{u_k\}_{k \ge 0}$ by iterating the following loop (see, e.g., [23])

$$\mathsf{SOLVE} \implies \mathsf{ESTIMATE} \implies \mathsf{MARK} \implies \mathsf{REFINE}. \tag{5.1}$$

At each iteration, the REFINE module either performs a local refinement of the underlying triangulation $\mathcal{T}$ or enriches the set $\mathfrak{P}$ of active indices. Two versions of the algorithm will be presented, which are different in the way the choice between mesh refinement and enrichment of the index set is made. Before discussing this aspect of the algorithm, let us describe the subroutines associated with four modules in (5.1). Throughout this section, we will use the subscript (or superscript) k $(k \ge 0)$ for triangulations, index sets, Galerkin solutions, *etc.* associated with the k-th iteration of the adaptive loop (5.1).

At each iteration $k \ge 0$, the finite element space $X(\mathcal{T}_k)$ associated with a conforming and shape-regular triangulation $\mathcal{T}_k$ is tensorized with the polynomial space $\mathcal{P}_{\mathfrak{P}_k}$. The unique sGFEM solution $u_k \in V_{X\mathfrak{P}}^{(k)} := X(\mathcal{T}_k) \otimes \mathcal{P}_{\mathfrak{P}_k}$ satisfying (3.5) is computed by the subroutine SOLVE as follows:

$$u_k = \texttt{SOLVE}\,(\mathcal{T}_k, \mathfrak{P}_k, a, f)\,,$$

where a and f are the problem data (see (2.1), (2.2)).

In order to control the error in the Galerkin solution u_k, local (spatial) estimators $\{e_{Y\mathfrak{P}}|_K\}_{K\in\mathcal{T}_k}$ and individual (parametric) estimators $\{e_{X\mathfrak{Q}}^{(\mu)}\}_{\mu\in\mathfrak{Q}_k}$ are computed as described in Section 4.1 by the subroutine ESTIMATE:

$$\Big[\{e_{Y\mathfrak{P}}|_K\}_{K\in\mathcal{T}_k}, \{e_{X\mathfrak{Q}}^{(\mu)}\}_{\mu\in\mathfrak{Q}_k}\Big] = \texttt{ESTIMATE}\big(u_k, \mathcal{T}_k, \mathfrak{P}_k, \mathfrak{Q}_k, a, f\big).$$

Here, the detail index set $\mathfrak{Q}_k \subset \mathfrak{I} \setminus \mathfrak{P}_k$ is built via (4.11). The total error estimate η_k is then calculated via (4.13).

If a prescribed tolerance ε is met by the Galerkin solution u_k (i.e., if $\eta_k < \varepsilon$), then the adaptive process stops. Otherwise, an enriched finite-dimensional subspace $V_{X\mathfrak{P}}^{(k+1)} \supset V_{X\mathfrak{P}}^{(k)}$ needs to be constructed and a more accurate Galerkin solution needs to be computed. At this stage in the adaptive loop, the module MARK identifies a subset $\mathcal{M}_k \subseteq \mathcal{T}_k$ of finite elements to be refined and a subset $\mathfrak{M}_k \subseteq \mathfrak{Q}_k$ of indices to be added to the current index set (although the marking is performed for both finite elements and indices, we recall that only one part of the approximation space $V_{X\mathfrak{P}}^{(k)}$ will be enriched afterwards). We employ the Dörfler strategy [9] for marking finite elements and indices. Specifically, we fix two threshold parameters θ_X, $\theta_{\mathfrak{P}} \in (0, 1]$ and build a minimal subset of marked elements $\mathcal{M}_k \subseteq \mathcal{T}_k$ satisfying

$$\sum_{K\in\mathcal{M}_k} \|e_{Y\mathfrak{P}}|_K\|_{B_0,K}^2 \geq \theta_X \sum_{K\in\mathcal{T}_k} \|e_{Y\mathfrak{P}}|_K\|_{B_0,K}^2, \tag{5.2}$$

as well as a minimal subset of marked indices $\mathfrak{M}_k \subseteq \mathfrak{Q}_k$ such that

$$\sum_{\mu\in\mathfrak{M}_k} \|e_{X\mathfrak{Q}}^{(\mu)}\|_{B_0}^2 \geq \theta_{\mathfrak{P}} \sum_{\mu\in\mathfrak{Q}_k} \|e_{X\mathfrak{Q}}^{(\mu)}\|_{B_0}^2.$$

As usual for Dörfler marking, larger values of the threshold parameter lead to larger subsets of marked elements (resp., indices), and it is guaranteed that sufficiently many elements (resp., indices) are selected so that their combined contribution to the total spatial (resp., parametric) error estimate constitutes a fixed proportion thereof. In the algorithm, the subsets $\mathcal{M}_k$ and $\mathfrak{M}_k$ are returned by the same subroutine MARK:

$$\mathcal{M}_k = \texttt{MARK}\big(\{\|e_{Y\mathfrak{P}}|_K\|_{B_0,K}\}_{K\in\mathcal{T}_k}, \theta_X\big), \quad \mathfrak{M}_k = \texttt{MARK}\big(\{\|e_{X\mathfrak{Q}}^{(\mu)}\|_{B_0}\}_{\mu\in\mathfrak{Q}_k}, \theta_{\mathfrak{P}}\big). \tag{5.3}$$

At this point in the adaptive loop, the algorithm needs to choose whether to enrich the finite element space or the polynomial space. Then, based on the chosen enrichment type and given the output of the subroutine MARK, the module REFINE in (5.1) either performs refinement of the current triangulation or enriches the current index set.

While polynomial enrichment is simply made by adding all marked indices to the current index set, i.e., by setting

$$\mathfrak{P}_{k+1} = \mathfrak{P}_k \cup \mathfrak{M}_k,$$

a refinement rule has to be set up in order to obtain a refined triangulation of $\mathcal{T}_k$. At the k-th iteration of the adaptive loop, a refined triangulation is returned by the subroutine

$$\mathcal{T}_{k+1} = \texttt{REFINE}(\mathcal{T}_k, \mathcal{M}_k) \tag{5.4}$$

implementing the *Longest Edge Bisection* (LEB) strategy [21] with a recursive edge-marking procedure that ensures the conformity of the triangulation $\mathcal{T}_{k+1}$. The LEB strategy is a variant of the Newest Vertex Bisection (NVB) algorithm (see, e.g., [20, 4, 18, 24]). We will denote by $\mathcal{R}_k \subseteq \mathcal{T}_k$ the set of all refined elements at the k-th iteration of the adaptive loop. Note that $\mathcal{R}_k \supseteq \mathcal{M}_k$. It is also easy to see that $\mathcal{R}_k = \mathcal{T}_k \setminus \mathcal{T}_{k+1}$, where $\mathcal{T}_{k+1}$ is the output of the subroutine `REFINE` in (5.4).

Let us now describe how the algorithm chooses between mesh refinement and polynomial enrichment at each iteration of the adaptive loop (5.1). In this respect, we distinguish two versions of the adaptive algorithm.

Version 1. In this version of the algorithm, we compare the *overall* spatial and parametric error estimates,

$$\eta_{\mathcal{T}} := \left(\sum_{K \in \mathcal{T}_k} \left\| e_{Y\mathfrak{P}}|_K \right\|^2_{B_{0,K}} \right)^{1/2} \quad \text{and} \quad \eta_{\mathfrak{Q}} := \left(\sum_{\mu \in \mathfrak{Q}_k} \| e^{(\mu)}_{X\mathfrak{Q}} \|^2_{B_0} \right)^{1/2}, \tag{5.5}$$

respectively, that contribute to the total error estimate η_k (see (4.13)).

If $\eta_{\mathcal{T}} \geq \eta_{\mathfrak{Q}}$, then local mesh refinement is performed and a new finite element space will be constructed at the next iteration of the algorithm, while the polynomial space will stay unchanged. Otherwise, i.e., if $\eta_{\mathfrak{Q}} > \eta_{\mathcal{T}}$, the index set is enriched and, at the next iteration of the algorithm, an enriched polynomial space will be formed, while the finite element space will stay unchanged.

The idea to compare two contributions to the total error estimate in order to decide on the enrichment type is not new. On the one hand, this idea was used in the adaptive algorithms described in [10, 11] and [12], where residual-based and local equilibration error estimators were employed. On the other hand, it was used in the adaptive algorithm with uniform mesh refinement presented in [7]. Note, however, that the estimate $\eta_{\mathcal{T}}$ combining all elementwise contributions $\{ \| e_{Y\mathfrak{P}}|_K \|^2_{B_{0,K}} \}_{K \in \mathcal{T}_k}$ does not necessarily provide an effective estimate of the error reduction that would be achieved if only the elements in the set $\mathcal{R}_k \subseteq \mathcal{T}_k$ were refined. Likewise, $\eta_{\mathfrak{Q}}$ does not necessarily estimate the error reduction that would be achieved by adding the *marked* indices $\mathfrak{M}_k \subset \mathfrak{Q}_k$. These observations motivate the second version of our adaptive algorithm.

Version 2. In this version, before deciding on the type of enrichment, we run the subroutine `REFINE` in (5.4) in order to obtain the set $\mathcal{R}_k$ of all elements that would be refined in case the mesh refinement is chosen. With the detail index set $\mathfrak{M}_k$ of marked indices already in our hands (see (5.3)), we consider the following two quantities:

$$\eta_{\mathcal{R}} := \left(\sum_{K \in \mathcal{R}_k} \left\| e_{Y\mathfrak{P}}|_K \right\|^2_{B_{0,K}} \right)^{1/2} \quad \text{and} \quad \eta_{\mathfrak{M}} := \left(\sum_{\mu \in \mathfrak{M}_k} \| e^{(\mu)}_{X\mathfrak{Q}} \|^2_{B_0} \right)^{1/2}. \tag{5.6}$$

Note that $\eta_{\mathcal{R}} \leq \eta_{\mathcal{T}}$ and $\eta_{\mathfrak{M}} \leq \eta_{\mathfrak{Q}}$, where $\eta_{\mathcal{T}}$ and $\eta_{\mathfrak{Q}}$ are defined in (5.5). Furthermore, as summation in (5.6) is over the elements to be refined (resp., over the marked indices to be added to the current index set), the quantity $\eta_{\mathcal{R}}$ (resp., $\eta_{\mathfrak{M}}$) does provide an effective estimate of the error reduction that would be achieved as a result of mesh refinement

```
Input: data $a$, $f$; coarse mesh $\mathcal{T}_0$, initial index set $\mathfrak{P}_0$;
       marking thresholds $\theta_X$, $\theta_{\mathfrak{P}}$; tolerance $\varepsilon$;
Output: final Galerkin solution $u_k$, final energy error estimate $\eta_k$;
for $k = 0, 1, 2, \ldots$ do
    $u_k =$ SOLVE($\mathcal{T}_k, \mathfrak{P}_k, a, f$);
    $\big[\{e_{Y\mathfrak{P}}|_K\}_{K\in\mathcal{T}_k}, \{e^{(\mu)}_{X\mathfrak{Q}}\}_{\mu\in\mathfrak{Q}_k}\big] =$ ESTIMATE $(u_k, \mathcal{T}_k, \mathfrak{P}_k, \mathfrak{Q}_k, a, f)$;
    $\eta_k = \Big(\sum_{K\in\mathcal{T}_k} \|e_{Y\mathfrak{P}}|_K\|^2_{B_{0,K}} + \sum_{\mu\in\mathfrak{Q}_k} \|e^{(\mu)}_{X\mathfrak{Q}}\|^2_{B_0}\Big)^{1/2}$;
    if $\eta_k < \varepsilon$ then break;
    $\mathcal{M}_k =$ MARK($\{\|e_{Y\mathfrak{P}}|_K\|_{B_{0,K}}\}_{K\in\mathcal{T}_k}, \theta_X$);
    $\mathfrak{M}_k =$ MARK($\{\|e^{(\mu)}_{X\mathfrak{Q}}\|_{B_0}\}_{\mu\in\mathfrak{Q}_k}, \theta_{\mathfrak{P}}$);
    % at this point, one of the two versions of the algorithm will run
    if Version 1 then
        $\eta_{\mathcal{T}} = (\sum_{K\in\mathcal{T}_k} \|e_{Y\mathfrak{P}}|_K\|^2_{B_{0,K}})^{1/2}$;  $\eta_{\mathfrak{Q}} = (\sum_{\mu\in\mathfrak{Q}_k} \|e^{(\mu)}_{X\mathfrak{Q}}\|^2_{B_0})^{1/2}$;
        if $\eta_{\mathcal{T}} \ge \eta_{\mathfrak{Q}}$ then $\mathcal{T}_{k+1} =$ REFINE($\mathcal{T}_k, \mathcal{M}_k$);  $\mathfrak{P}_{k+1} = \mathfrak{P}_k$;
        else $\mathcal{T}_{k+1} = \mathcal{T}_k$;  $\mathfrak{P}_{k+1} = \mathfrak{P}_k \cup \mathfrak{M}_k$;
    else % if Version 2
        $\mathcal{T} =$ REFINE($\mathcal{T}_k, \mathcal{M}_k$);   $\mathcal{R}_k = \mathcal{T}_k \setminus \mathcal{T}$;
        $\eta_{\mathcal{R}} = (\sum_{K\in\mathcal{R}_k} \|e_{Y\mathfrak{P}}|_K\|^2_{B_{0,K}})^{1/2}$;  $\eta_{\mathfrak{M}} = (\sum_{\mu\in\mathfrak{M}_k} \|e^{(\mu)}_{X\mathfrak{Q}}\|^2_{B_0})^{1/2}$;
        if $\eta_{\mathcal{R}} \ge \eta_{\mathfrak{M}}$ then $\mathcal{T}_{k+1} = \mathcal{T}$;  $\mathfrak{P}_{k+1} = \mathfrak{P}_k$;
        else $\mathcal{T}_{k+1} = \mathcal{T}_k$;  $\mathfrak{P}_{k+1} = \mathfrak{P}_k \cup \mathfrak{M}_k$;
    end
end
```

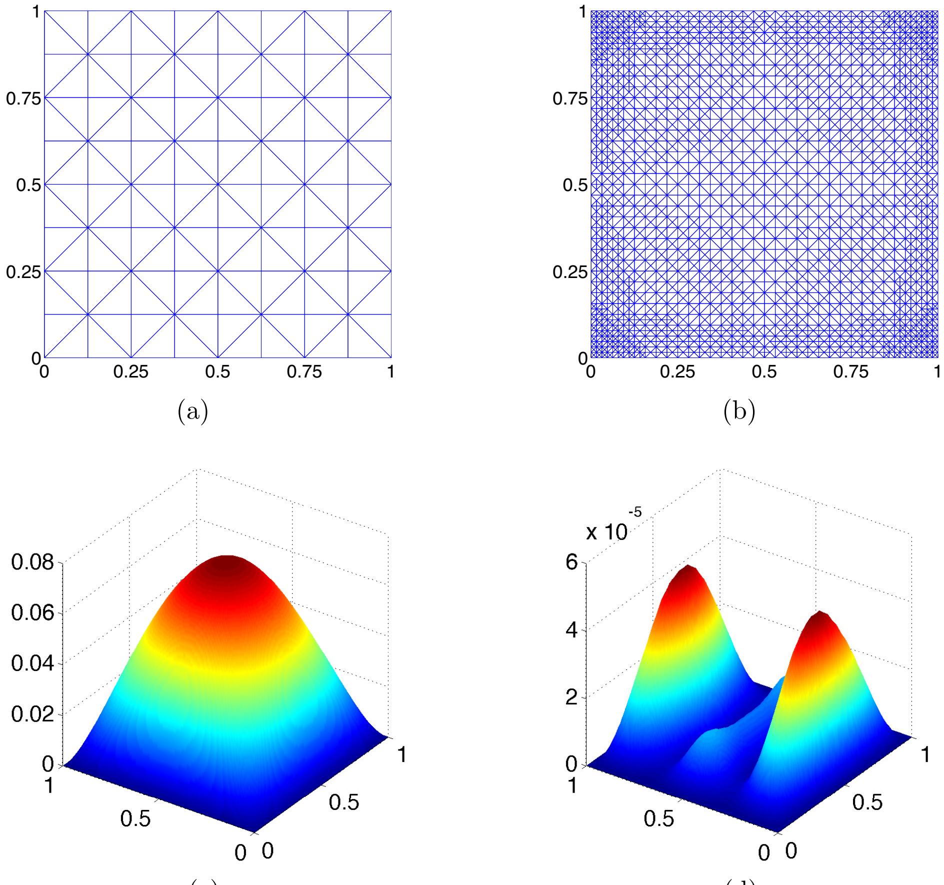
Algorithm 5.1. Adaptive stochastic Galerkin finite element algorithm.

(resp., polynomial enrichment), cf. Proposition 4.2. Therefore, in the spirit of algorithms driven by dominant error reduction estimates, the enrichment type in this version is chosen by comparing the quantities $\eta_{\mathcal{R}}$ and $\eta_{\mathfrak{M}}$ in (5.6). More precisely, if $\eta_{\mathcal{R}} \ge \eta_{\mathfrak{M}}$, then the mesh refinement is performed, otherwise, the index set is enriched.

The complete adaptive algorithm incorporating the two versions described above is listed in Algorithm 5.1.

6 Numerical experiments

Let us report the results of some numerical experiments that were performed for the parametric model problem (2.1). These results illustrate some aspects of the design of adaptive algorithms for parametric PDEs and demonstrate the performance of two versions of the adaptive algorithm described in Section 5. All numerical experiments were performed using the open source Matlab toolbox Stochastic T-IFISS [6] on a desktop



(a)

(b)

(c)

(d)

Figure 6.1. (a) Initial coarse triangulation $\mathcal{T}_0$ in Experiment 1; (b) adaptively refined triangulation produced by Version 1 of Algorithm 5.1 in Experiment 1; (c)–(d) the mean field $\mathbb{E}[u_{X\mathfrak{P}}]$ and the variance $\mathrm{Var}(u_{X\mathfrak{P}})$ of the computed sGFEM solution for the model problem in Experiment 1.

computer equipped with an Intel Core CPU i5-4590@3.30GHz and 8.00GB RAM. In all experiments, when running Algorithm 5.1 we use the initial index set given by

$$\mathfrak{P}_0 := \{(0,0,0,\dots),\,(1,0,0,\dots)\}. \tag{6.1}$$

Experiment 1. In the first experiment, we demonstrate the performance of two versions of Algorithm 5.1 for the parametric model problem (2.1) posed on the square domain $D = (0,1)^2$. We follow Eigel et al. [10, Section 11.1] and choose the expansion coefficients $a_m(\mathbf{x})$, $m \in \mathbb{N}_0$ in (2.2) to represent planar Fourier modes of increasing total order. Specifically, we set

$$a_0(\mathbf{x}) = 1, \quad a_m(\mathbf{x}) = \bar{\alpha} m^{-\tilde{\sigma}} \cos(2\pi\beta_1(m)\,x_1)\cos(2\pi\beta_2(m)\,x_2), \quad \mathbf{x} \in D. \tag{6.2}$$

Here,

$$\beta_1(m) = m - k(m)(k(m)+1)/2 \;\text{ and }\; \beta_2(m) = k(m) - \beta_1(m)$$

with $k(m) = \left\lfloor -1/2 + \sqrt{1/4 + 2m} \right\rfloor$, $\tilde{\sigma} > 1$ and $0 < \bar{\alpha} < 1/\zeta(\tilde{\sigma})$, where ζ denotes the Riemann zeta function. Note that with this choice of expansion coefficients, the weak formulation (2.11) always admits a unique solution $u \in V$. This is because $\alpha_0^{\min} = \alpha_0^{\max} = 1$ in (2.3) and $\tau = \bar{\alpha}\zeta(\tilde{\sigma}) < 1$ as required by (2.4). In particular, setting $\tilde{\sigma} = 2$ in (6.2),

	Case (i): $\theta_X = 0.5,\ \theta_{\mathfrak{P}} = 0.9$		Case (ii): $\theta_X = 0.2,\ \theta_{\mathfrak{P}} = 0.9$	
	Version 1	Version 2	Version 1	Version 2
t, sec	395	289	605	479
K	27	26	64	61
η_K	1.2652e-03	1.4438e-03	1.2509e-03	1.4369e-03
$\#\mathcal{T}_K$	87‘520	65‘750	86‘552	64‘156
$\#\mathcal{N}_K$	43‘381	32‘546	42‘903	31‘753
$\#\mathfrak{P}_K$	23	23	23	23
N_K	997‘763	748‘558	986‘769	730‘319
$\mathfrak{P}$	$k = 0$ (0 0) (1 0)	$k = 0$ (0 0) (1 0)	$k = 0$ (0 0) (1 0)	$k = 0$ (0 0) (1 0)
	$k = 9$ (0 1) (2 0)	$k = 8$ (0 1) (2 0)	$k = 21$ (0 1) (2 0)	$k = 10$ (0 1) (2 0)
	$k = 15$ (0 0 1) (1 1 0) (3 0 0)	$k = 13$ (0 0 1) (1 1 0) (3 0 0)	$k = 35$ (0 0 1) (1 1 0) (3 0 0)	$k = 23$ (0 0 1) (1 1 0) (3 0 0)
	$k = 20$ (0 0 0 1) (1 0 1 0) (2 1 0 0)	$k = 19$ (0 0 0 1) (1 0 1 0) (2 1 0 0)	$k = 48$ (0 0 0 1) (1 0 1 0) (2 1 0 0)	$k = 36$ (0 0 0 1) (1 0 1 0) (2 1 0 0)
	$k = 24$ (0 0 0 0 1) (0 2 0 0 0) (1 0 0 1 0) (2 0 1 0 0) (3 1 0 0 0) (4 0 0 0 0)	$k = 22$ (0 0 0 0 1) (0 2 0 0 0) (1 0 0 1 0) (2 0 1 0 0) (3 1 0 0 0) (4 0 0 0 0)	$k = 56$ (0 0 0 0 1) (0 2 0 0 0) (1 0 0 1 0) (2 0 1 0 0) (3 1 0 0 0) (4 0 0 0 0)	$k = 44$ (0 0 0 0 1) (0 2 0 0 0) (1 0 0 1 0) (2 0 1 0 0) (3 1 0 0 0) (4 0 0 0 0)
	$k = 27$ (0 0 0 0 0 1) (0 1 1 0 0 0) (1 0 0 0 0 1) (1 0 0 0 1 0) (1 2 0 0 0 0) (2 0 0 1 0 0) (3 0 1 0 0 0)	$k = 26$ (0 0 0 0 0 1) (0 1 1 0 0 0) (1 0 0 0 0 1) (1 0 0 0 1 0) (1 2 0 0 0 0) (2 0 0 1 0 0) (3 0 1 0 0 0)	$k = 64$ (0 0 0 0 0 1) (0 1 1 0 0 0) (1 0 0 0 0 1) (1 0 0 0 1 0) (1 2 0 0 0 0) (2 0 0 1 0 0) (3 0 1 0 0 0)	$k = 51$ (0 0 0 0 0 1) (0 1 1 0 0 0) (1 0 0 0 0 1) (1 0 0 0 1 0) (1 2 0 0 0 0) (2 0 0 1 0 0) (3 0 1 0 0 0)

Table 6.1. The results of running two versions of Algorithm 5.1 with two sets of Dörfler marking parameters for the model problem in Experiment 1.

we select $\bar{\alpha}$ such that $\tau = \bar{\alpha}\zeta(\tilde{\sigma}) = 0.9$. This choice of parameters corresponds to a slow decay of the amplitudes $\bar{\alpha}m^{-\tilde{\sigma}}$ and gives $\bar{\alpha} \approx 0.547$ (cf. [10, Section 11.1]). Furthermore, we set $f(\mathbf{x}) = 1$ for all $\mathbf{x} = (x_1, x_2) \in D$ and assume that the parameters y_m in (2.2) are the images of uniformly distributed independent mean-zero random variables, so that $\pi_m = \pi_m(y_m)$ is the associated probability measure on $\Gamma_m = [-1, 1]$ and $d\pi_m = \frac{1}{2}dy_m$. The same model problem as described above has been used in numerical experiments in [10, 11, 7, 12, 13].

We run Version 1 and Version 2 of Algorithm 5.1 with the same sets of input parameters and data. More precisely, we use the initial (coarse) triangulation $\mathcal{T}_0$ depicted in Figure 6.1(a) and the initial index set $\mathfrak{P}_0$ given by (6.1). For marking purposes, we use two sets of Dörfler marking parameters: (i) $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.9$; (ii) $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.9$. The same stopping tolerance $\varepsilon = 1.5\text{e-}3$ is set in all cases. The results of these computations are presented in Table 6.1 and in Figures 6.1, 6.2, and 6.3.

Figure 6.1(b) shows the locally refined triangulation produced by Version 1 of the adaptive algorithm in case (i) when an intermediate tolerance equal to 7.0e-3 was met

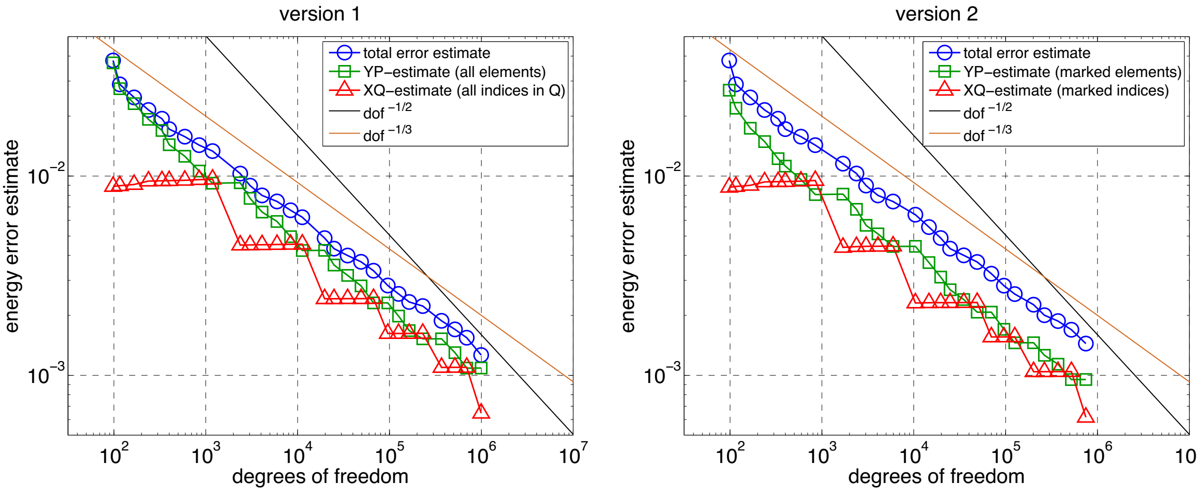


Figure 6.2. Energy error estimates at each step of the adaptive algorithm with $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.9$ (case (i)) for the model problem in Experiment 1.

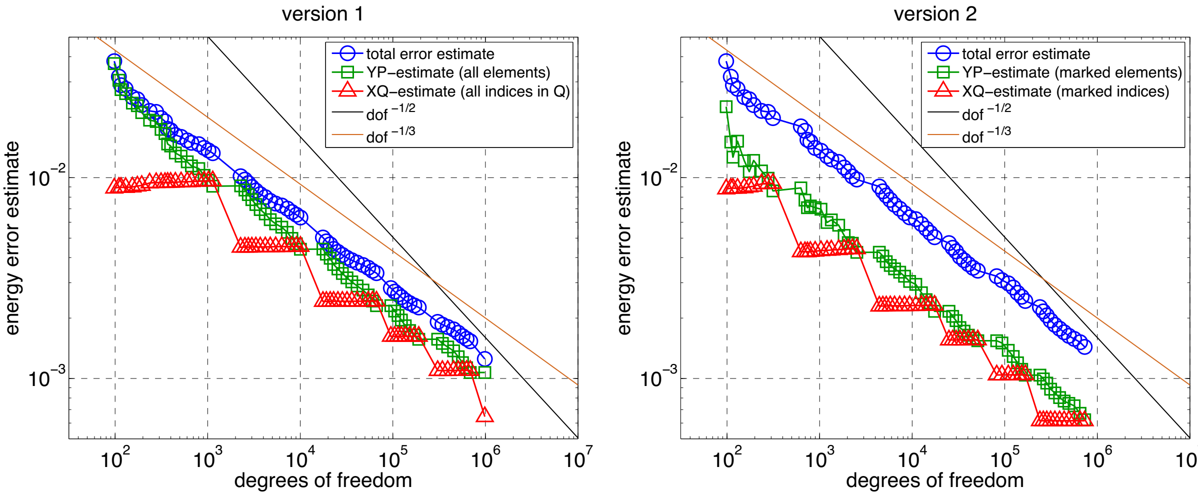


Figure 6.3. Energy error estimates at each step of the adaptive algorithm with $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.9$ (case (ii)) for the model problem in Experiment 1.

(similar triangulations were produced in all other cases). Figure 6.1(c)–(d) shows the mean and the variance of the computed sGFEM solution. Note that due to the regularity of the solution and since the magnitude of the variance is much smaller than the magnitude of the mean field, the triangulation is refined towards the corners of the domain.

In Table 6.1, for each computation we show the computational time t (in seconds), the total number of iterations K, the final error estimate η_K, the number of finite elements and the number of interior vertices of the final mesh $\mathcal{T}_K$, the cardinality of the final index set $\mathfrak{P}_K$, as well as the evolution of the index set $\mathfrak{P}$.

By looking at the results in Table 6.1, we observe some differences in the performance of two versions in terms of computational times, the final number of elements, and the total number of degrees of freedom (cf. the values of t, $\#\mathcal{T}_K$, and N_K in Table 6.1). In particular, Version 2 took less iterations and reached the tolerance faster than Version 1 (about 27% of time saved in case (i)). In case (i) and case (ii), both versions produced the same final index set $\mathfrak{P}_K$ with 23 indices corresponding to polynomials of total degree 4 in 6 active parameters. We also note that by design, Version 2 triggers polynomial enrichments at earlier iterations than Version 1. This results in a balanced refinement

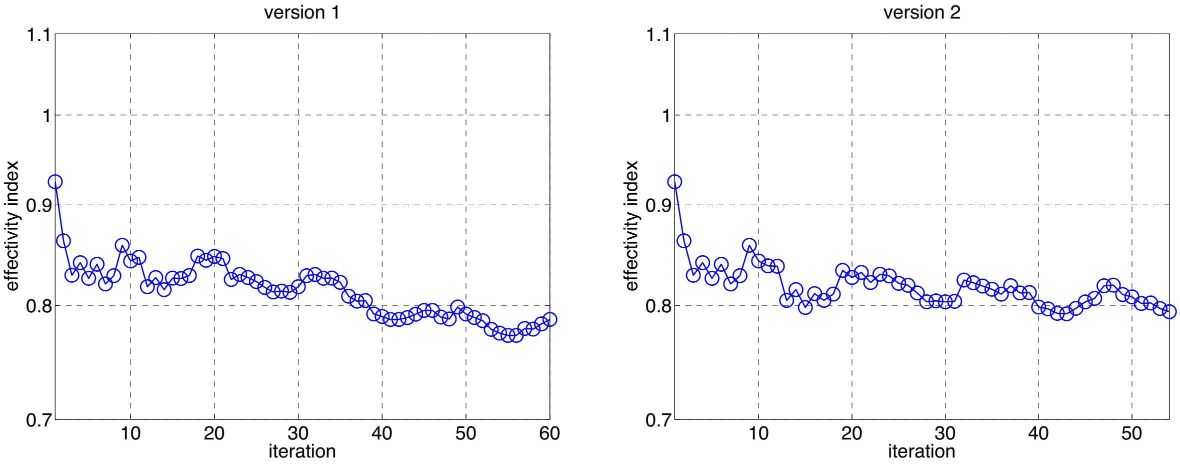


Figure 6.4. The effectivity indices for the sGFEM solutions in Experiment 1 at each iteration of the adaptive algorithm with $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.9$ (case (ii)).

of spatial and parametric components of Galerkin approximations generated by running Version 2 and is one of the reasons why Version 2 is faster and overall more efficient than Version 1 for the model problem in this experiment.

By looking now at Figures 6.2 and 6.3 we observe that the total error estimates decay with an overall rate of about $\mathcal{O}(N^{-1/3})$ for both versions and for both sets of marking thresholds. However, due to spatial regularity of the solution, we expect the error estimates to decay with the optimal rate $\mathcal{O}(N^{-1/2})$ during mesh refinement steps, cf. [11] (mesh refinements can be identified on the graphs as the steps where $Y\mathfrak{P}$-error estimates decay). It turns out that neither version achieves this optimal decay rate during mesh refinement stages in case (i) ($\theta_X = 0.5$), see Figure 6.2 (also cf. [11, Figure 1] in the case $\theta_X = 0.4$ and [7, Figure 1] for the case of uniform mesh refinement ($\theta_X = 1$)). However, in case (ii) ($\theta_X = 0.2$), Figure 6.3 shows that the decay rate during mesh refinement steps is very close to the optimal one for Version 2, while it is still far from being optimal for Version 1. We also note that, since polynomial enrichments are triggered earlier by Version 2, the associated reductions in the total error estimates during these steps are smaller than the error reductions that occur during polynomial enrichment steps when running Version 1.

We conclude this experiment by testing the effectivity of the error estimation at each step of the adaptive algorithm. To that end, we compare the error estimates η_k with the energy norm of a reference error $e_k^{\mathrm{ref}} := u_{\mathrm{ref}} - u_k$, where $u_{\mathrm{ref}} \in V_{X\mathfrak{P}}^{\mathrm{ref}} := X_{\mathrm{ref}} \otimes \mathcal{P}_{\mathfrak{P}_{\mathrm{ref}}}$ is an accurate (reference) solution. Using Galerkin orthogonality and the symmetry of the bilinear form B, we have

$$\|e_k^{\mathrm{ref}}\|_B = (\|u_{\mathrm{ref}}\|_B^2 - \|u_k\|_B^2)^{1/2},$$

and then the effectivity indices are computed as follows:

$$\theta_k = \eta_k \,/\, \|e_k^{\mathrm{ref}}\|_B, \quad \forall\, k \geq 0. \tag{6.3}$$

For the model problem in this experiment, we use the reference Galerkin solution u_{ref} from [7, Section 6]. The effectivity indices for both versions of the adaptive algorithm for case (ii) are plotted in Figure 6.4 (the plots are very similar in case (i)). We can see that the effectivity indices are less than unity throughout all iterations and tend to be close to 0.8 as iterations progress.

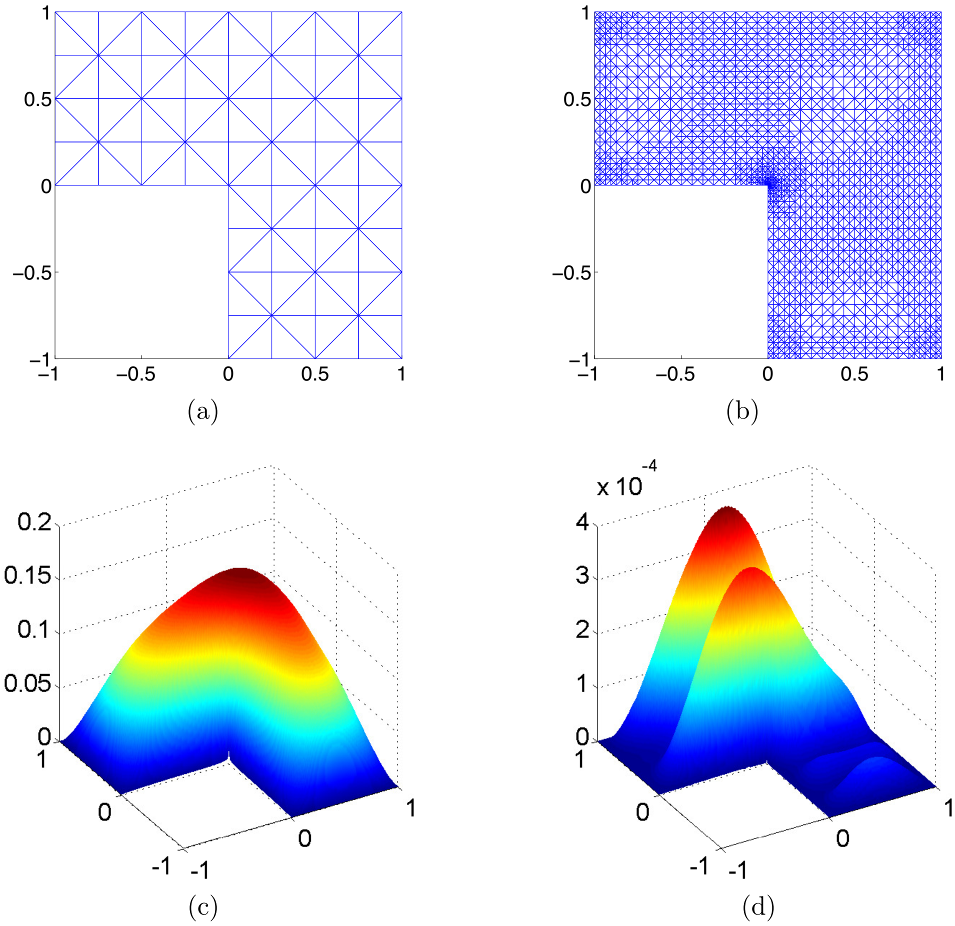


Figure 6.5. (a) Initial coarse triangulation $\mathcal{T}_0$ in Experiment 2; (b) adaptively refined triangulation produced by Version 2 of Algorithm 5.1 in Experiment 2; (c)–(d) the mean field $\mathbb{E}[u_{X\mathfrak{P}}]$ and the variance $\mathrm{Var}(u_{X\mathfrak{P}})$ of the computed sGFEM solution for the model problem in Experiment 2.

Based on the results of Experiment 1, we conclude that Version 2 of the adaptive algorithm is more efficient than Version 1 for the considered parametric problem on the square domain. Indeed, Version 2 reaches the desired tolerance faster and with a fewer number of total degrees of freedom; furthermore, the corresponding total error estimates decay with an optimal rate during mesh refinement steps, provided that the spatial marking threshold θ_X is sufficiently small (our experiments suggest to chose $\theta_X = 0.2$). On the other hand, the overall convergence rate is essentially the same for both versions of the algorithm and for both sets of marking parameters considered in this experiment.

Experiment 2. In the second experiment, we compare the performance of two versions of Algorithm 5.1 for the same parametric model problem (2.1) as in Experiment 1 but now posed on the L-shaped domain $D = (-1,1)^2 \setminus (-1,0]^2$. Exactly the same model problem has been solved numerically in [10, 11, 12, 13].

We use the initial (coarse) triangulation $\mathcal{T}_0$ depicted in Figure 6.5(a) and the initial index set $\mathfrak{P}_0$ given by (6.1). For marking purposes, we use two sets of Dörfler marking parameters: (i) $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.8$; (ii) $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.8$. The same stopping tolerance $\varepsilon = 5.0\text{e-}3$ is set in all cases. The results of these computations are presented in Table 6.2 and in Figures 6.5, 6.6, and 6.7.

Figure 6.5(b) shows the locally refined triangulation produced by Version 2 of Algo-

	Case (i): $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.8$		Case (ii): $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.8$	
	Version 1	Version 2	Version 1	Version 2
t, sec	371	395	476	419
K	27	27	65	63
η_K	4.7170e-03	4.7160e-03	4.8340e-03	4.9659e-03
$\#\mathcal{T}_K$	103‘206	103‘304	89‘480	67‘770
$\#\mathcal{N}_K$	51‘133	51‘182	44‘317	33‘533
$\#\mathfrak{P}_K$	13	13	13	18
N_K	664‘729	665‘366	576‘121	603‘594
$\mathfrak{P}$	$k = 0$ (0 0) (1 0)	$k = 0$ (0 0) (1 0)	$k = 0$ (0 0) (1 0)	$k = 0$ (0 0) (1 0)
	$k = 12$ (0 1) (2 0)	$k = 11$ (0 1) (2 0)	$k = 29$ (0 1) (2 0)	$k = 20$ (0 1) (2 0)
	$k = 19$ (0 0 1) (1 1 0)	$k = 17$ (0 0 1) (1 1 0)	$k = 45$ (0 0 1) (1 1 0)	$k = 35$ (0 0 1) (1 1 0)
	$k = 22$ (0 0 0 1) (1 0 1 0) (3 0 0 0)	$k = 21$ (0 0 0 1) (1 0 1 0) (3 0 0 0)	$k = 53$ (0 0 0 1) (1 0 1 0) (3 0 0 0)	$k = 43$ (0 0 0 1) (1 0 1 0) (3 0 0 0)
	$k = 26$ (0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (2 1 0 0 0)	$k = 25$ (0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (2 1 0 0 0)	$k = 62$ (0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (2 1 0 0 0)	$k = 52$ (0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (2 1 0 0 0)
				$k = 60$ (0 0 0 0 0 1) (0 2 0 0 0 0) (1 0 0 0 1 0) (3 1 0 0 0 0) (4 0 0 0 0 0)

Table 6.2. The results of running two versions of Algorithm 5.1 with two sets of Dörfler marking parameters for the model problem in Experiment 2.

rithm 5.1 with $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.8$ (case (ii)) when an intermediate tolerance 2.5e-2 was met (triangulations with a similar pattern were produced in all other cases). In Figure 6.5(c)–(d), the mean and the variance of the computed sGFEM solution are depicted. We observe that the adaptively refined mesh effectively identifies the area of singular behavior of the mean field (in the vicinity of the reentrant corner), where we can see much stronger mesh refinement than in other areas of the domain. Note that, since the magnitude of the mean is much higher than the one for the variance, a ‘roughness’ of variance in some areas of the domain does not have a significant impact on mesh refinement in those areas.

Table 6.2 shows the final outputs of all computations in this experiment. By looking at the results for case (i) ($\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.8$), we do not observe significant differences between the approximations produced by two versions of the algorithm. Indeed, the tolerance was reached after the same number of iterations ($K = 27$), the same final index set (with 13 indices) was generated, and the number of elements in final triangulations was comparable for both versions. Also, both versions took nearly the same time to reach the tolerance, although Version 1 was slightly faster than Version 2.

In the case when $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.8$ (case (ii)), the differences between the two versions are evident. To start with, Version 1 needed two iterations more than Version 2 to reach the tolerance. Furthermore, Version 1 produced a more refined triangulation

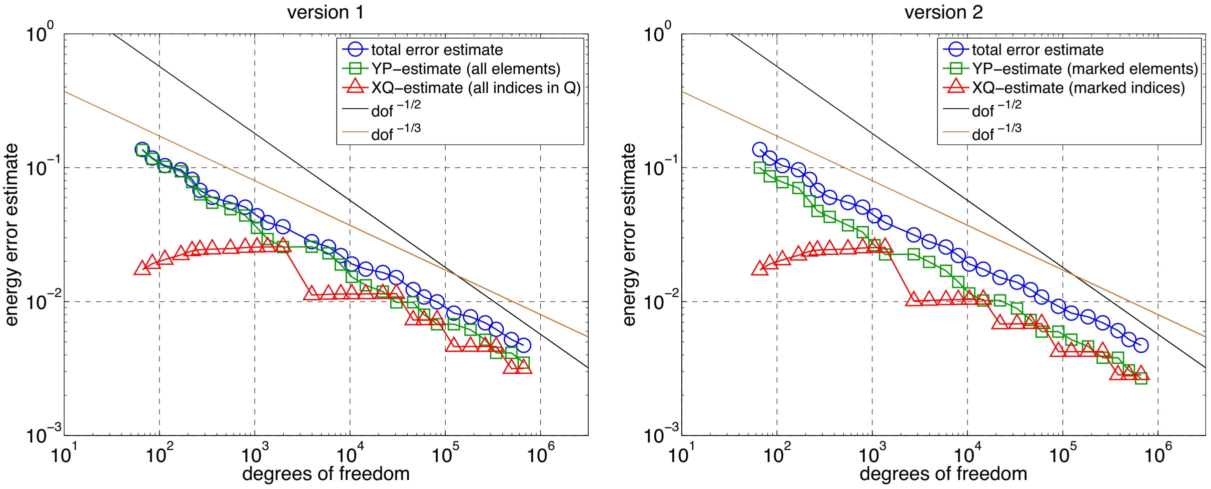


Figure 6.6. Energy error estimates at each step of the adaptive algorithm with $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.8$ (case (i)) for the model problem in Experiment 2.

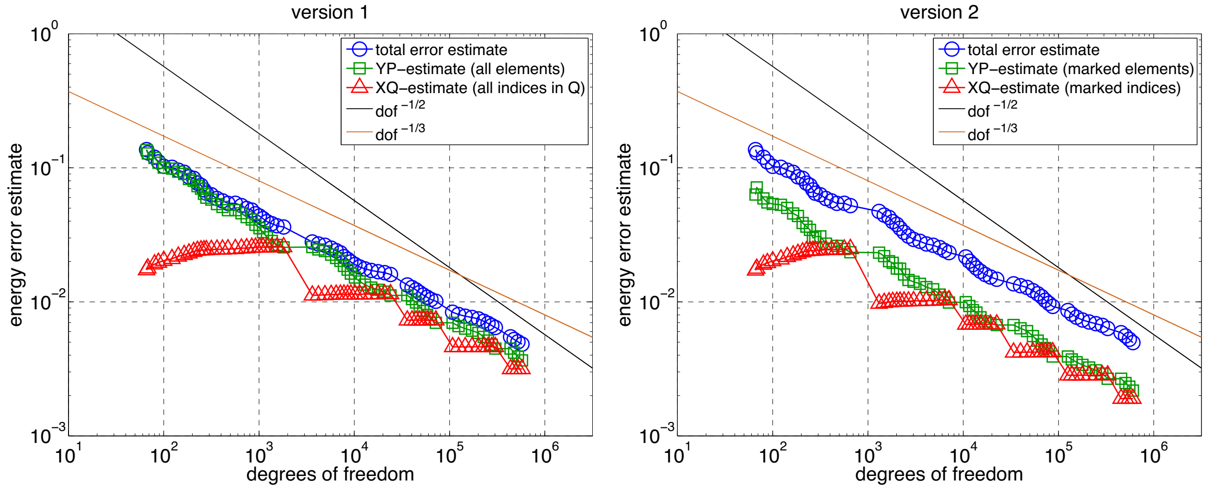


Figure 6.7. Energy error estimates at each step of the adaptive algorithm with $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.8$ (case (ii)) for the model problem in Experiment 2.

than Version 2 did (cf. the values of $\#\mathcal{T}_K$ in Table 6.2). On the other hand, Version 2 generated a more developed index set with more active parameters and higher degree of polynomial approximation in these parameters. This explains why Version 2 terminated with a slightly bigger number of total degrees of freedom in this case. More importantly, Version 2 was about 12% faster than Version 1; as already observed in Experiment 1, this was due to polynomial enrichments triggered at earlier iterations.

By looking now at Figures 6.6 and 6.7 we see that the overall convergence rate for the total error estimate is about $\mathcal{O}(N^{-1/3})$ for both versions and for both sets of marking thresholds. However, we expect the error estimates to decay with the optimal rate $\mathcal{O}(N^{-1/2})$ during mesh refinement steps, cf. [11]. The optimal rate is not achieved in case (i) due to the fact that the marking threshold $\theta_X = 0.5$ is not sufficiently small (see, e.g., [8, 9]). In case (ii), i.e., for $\theta_X = 0.2$, the decay rate during mesh refinement steps is close to the optimal one only for Version 2. This observation is consistent with the one made in Experiment 1 on the square domain.

Following the procedure described in Experiment 1, we now compute the effectivity indices θ_k given by (6.3) at each step of the algorithm. In particular, we employ the

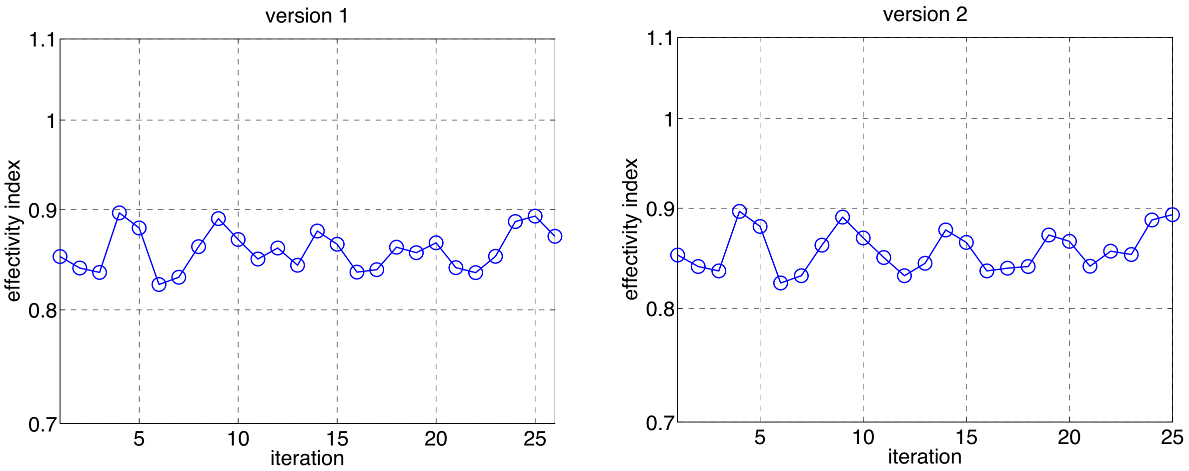


Figure 6.8. The effectivity indices for the sGFEM solutions in Experiment 2 at each iteration of the adaptive algorithm with $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.8$ (case (i)).

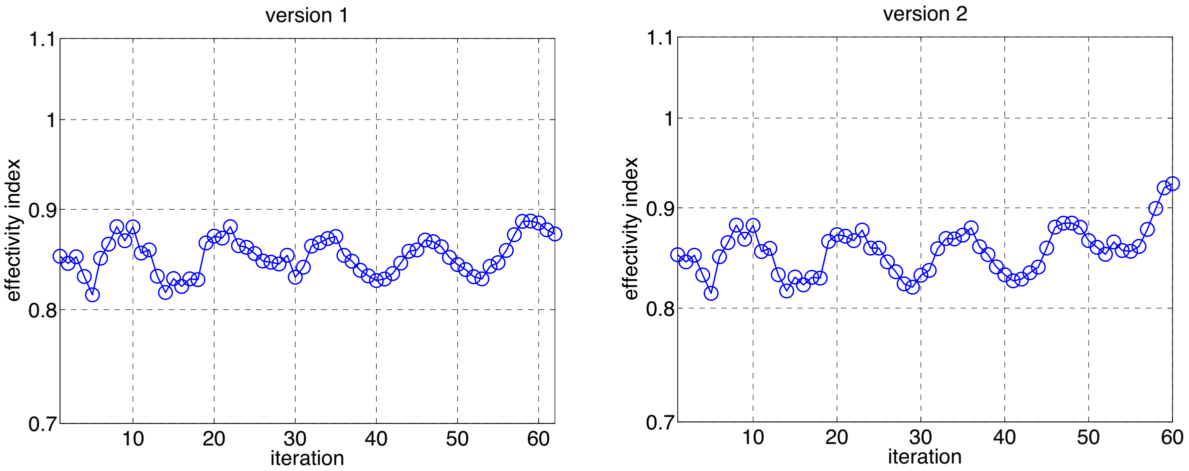


Figure 6.9. The effectivity indices for the sGFEM solutions in Experiment 2 at each iteration of the adaptive algorithm with $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.8$ (case (ii)).

reference Galerkin solution computed with quadratic (P_2) spatial approximations over a fine grid (the final triangulation produced by Version 2 in case (i) with an additional uniform refinement) and using a large index set (the final index set generated by Version 2 in case (ii)). The effectivity indices are plotted in Figure 6.8 (case (i)) and Figure 6.9 (case (ii)). In each case, the effectivity indices lie within the interval (0.8, 0.93) throughout all iterations.

Thus, in agreement with the results of Experiment 1, we conclude that for the parametric model problem on the L-shaped domain, Version 2 of Algorithm 5.1 is more efficient than Version 1, if the spatial marking threshold θ_X is sufficiently small. In particular, our experiments suggest to choose $\theta_X = 0.2$. In this case, Version 2 produces more accurate parametric approximations by generating a richer index set, and the associated total error estimates decay with an optimal rate during mesh refinement steps.

Experiment 3. In this experiment, we consider the parametric model problem (2.1) posed on the square domain with a crack, i.e.,

$$D = (-1,1)^2 \setminus \{(x_1,x_2) \in \mathbb{R}^2 \,:\, -1 < x_1 \leq 0,\ x_2 = 0\}.$$

We set $f(\mathbf{x}) = \exp(-(x_1+0.5)^2 - (x_2-0.5)^2)$ for all $\mathbf{x} = (x_1,x_2) \in D$ and consider the

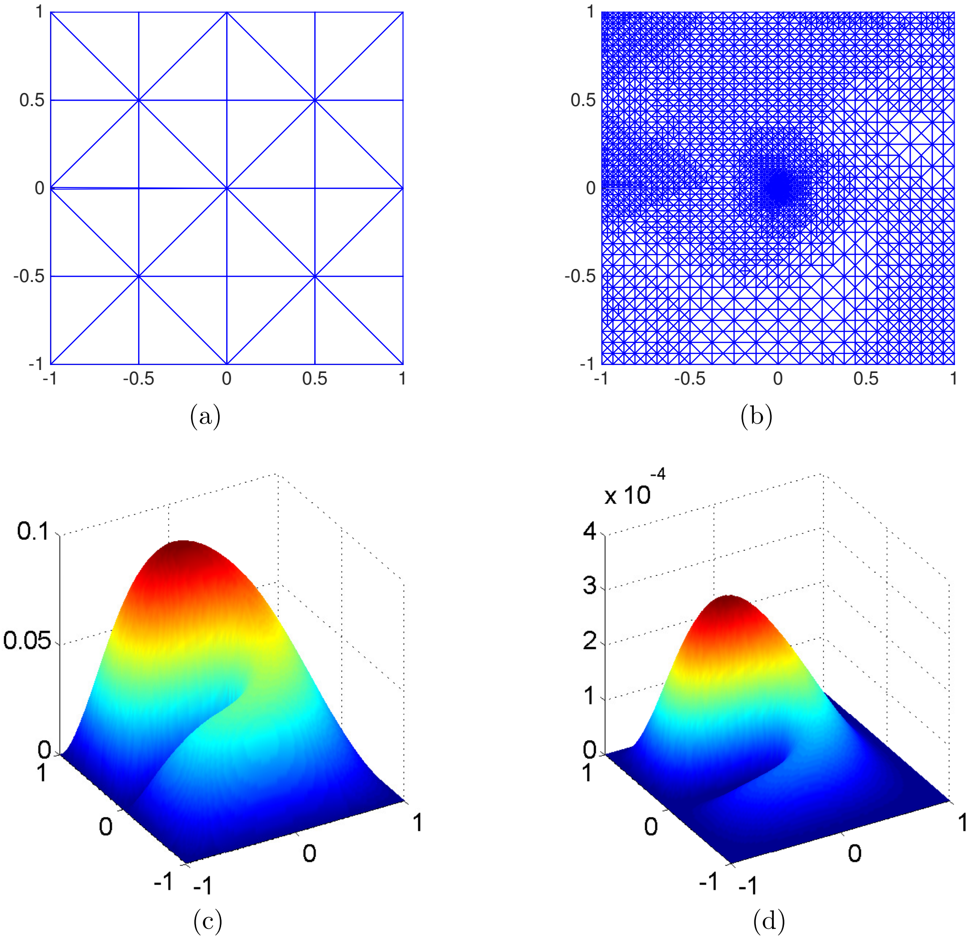


Figure 6.10. (a) Initial coarse triangulation $\mathcal{T}_0$ in Experiment 3; (b) adaptively refined triangulation produced by Version 2 of Algorithm 5.1 in Experiment 3; (c)–(d) the mean field $\mathbb{E}[u_{X\mathfrak{P}}]$ and the variance $\mathrm{Var}(u_{X\mathfrak{P}})$ of the computed sGFEM solution for the model problem in Experiment 3.

following parametric diffusion coefficient (cf. [19, Example 9.37])

$$a(\mathbf{x}, \mathbf{y}) = 1 + \frac{1}{\sqrt{3}} \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sqrt{\nu_{ij}} \phi_{ij}(\mathbf{x}) y_{ij}, \tag{6.4}$$

where $\phi_{00}(\mathbf{x}) := 1$, $\nu_{00} := 1/4$,

$$\phi_{ij}(\mathbf{x}) := 2\cos(i\pi x_1)\cos(j\pi x_2), \quad \nu_{ij} := \frac{1}{4} e^{-\pi(i^2+j^2)}, \quad i, j \geq 1,$$

and $y_{ij} \in [-1, 1]$ $(i, j \in \mathbb{N}_0)$ are the images of the uniformly distributed mean-zero random variables. We rewrite the sum in (6.4) in terms of a single index m so that the values ν_m appear in descending order:

$$a(\mathbf{x}, \mathbf{y}) = 1 + \frac{1}{\sqrt{3}} \sum_{m=1}^{\infty} \sqrt{\nu_m} \phi_m(\mathbf{x}) y_m, \qquad \mathbf{y} \in \Gamma.$$

For the model problem described above, we again run two versions of Algorithm 5.1 with the initial (coarse) triangulation $\mathcal{T}_0$ depicted in Figure 6.10(a) and the initial index

	Case (i): $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.9$		Case (ii): $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.9$		Case (iii): $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.5$	
t, sec	1321		1655		1723	
K	80		36		37	
η_K	1.9868e-03		1.8075e-03		1.8076e-03	
$\#\mathcal{T}_K$	213‘830		286‘915		286‘891	
$\#\mathcal{N}_K$	106‘240		142‘699		142‘687	
$\#\mathfrak{P}_K$	9		9		9	
N_K	956‘160		1‘284‘291		1‘284‘183	
$\mathfrak{P}$	$k = 0$	(0 0) (1 0)	$k = 0$	(0 0) (1 0)	$k = 0$	(0 0) (1 0)
	$k = 46$	(0 1) (2 0)	$k = 21$	(0 1) (2 0)	$k = 21$	(0 1) (2 0)
	$k = 51$	(0 0 1) (1 1 0)	$k = 24$	(0 0 1) (1 1 0)	$k = 24$	(0 0 1)
	$k = 68$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$k = 31$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$k = 30$	(1 0 1) (1 1 0)
					$k = 35$	(0 0 0 1) (3 0 0 0)

Table 6.3. The results of running Version 1 of Algorithm 5.1 with three sets of Dörfler marking parameters for the model problem in Experiment 3.

set $\mathfrak{P}_0$ given by (6.1). Aiming to understand the influence of both marking thresholds θ_X and $\theta_{\mathfrak{P}}$, we perform computations with three sets of Dörfler marking parameters:

$$\text{(i) } \theta_X = 0.2,\ \theta_{\mathfrak{P}} = 0.9; \quad \text{(ii) } \theta_X = 0.5,\ \theta_{\mathfrak{P}} = 0.9; \quad \text{(iii) } \theta_X = 0.5,\ \theta_{\mathfrak{P}} = 0.5.$$

The same stopping tolerance $\varepsilon = 2.0\text{e-}3$ was set in all computations. The results of these computations are presented in Tables 6.3, 6.4 and in Figures 6.10, 6.11.

Figure 6.10(b) shows the locally refined triangulation produced by Version 2 of Algorithm 5.1 with $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.9$ (case (ii)) when an intermediate tolerance 1.0e-2 was met. In Figure 6.10(c)–(d), the mean and the variance of the computed sGFEM solution are plotted. As in previous experiments, we see that the algorithm performs effective adaptive mesh refinement in the areas where the mean and the variance of the solution are not sufficiently smooth. For the model problem in this experiment, the strongest mesh refinement occurs in the vicinity of the crack tip.

By looking at the results in Tables 6.3 and 6.4, we can see that among six computations carried out in this experiment, the best performance in terms of computational time was achieved by Version 2 of the algorithm with $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.9$ (case (i)). In particular, it was 30% faster than Version 1 with the same marking thresholds. In agreement with the results of previous experiments, Version 2 produced less refined triangulations and triggered polynomial enrichment earlier than Version 1 in all three cases. Furthermore, in case (i), Version 2 needed two iterations less to reach the set tolerance, and the final index set was more developed than the index set generated by Version 1 in this case (15 versus 9 indices). The advantages of using a smaller marking threshold θ_X are again more evident when running Version 2: the final triangulation in case (i) has nearly twice less elements than in cases (ii) and (iii); the index set is more developed in case (i) than in

	Case (i): $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.9$		Case (ii): $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.9$		Case (iii): $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.5$	
t, sec	935		1811		1841	
K	78		36		37	
η_K	1.9591e-03		1.8068e-03		1.8069e-03	
$\#\mathcal{T}_K$	153‘312		287‘256		287‘212	
$\#\mathcal{N}_K$	760‘48		142‘869		142‘847	
$\#\mathfrak{P}_K$	15		9		9	
N_K	1‘140‘720		1‘285‘821		1‘285‘623	
$\mathfrak{P}$	$k = 0$	(0 0) (1 0)	$k = 0$	(0 0) (1 0)	$k = 0$	(0 0) (1 0)
	$k = 36$	(0 1) (2 0)	$k = 19$	(0 1) (2 0)	$k = 19$	(0 1) (2 0)
	$k = 41$	(0 0 1) (1 1 0)	$k = 22$	(0 0 1) (1 1 0)	$k = 22$	(0 0 1)
	$k = 57$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$k = 29$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$k = 28$	(1 0 1) (1 1 0)
	$k = 76$	(0 0 2 0) (0 1 1 0) (0 2 0 0) (1 0 0 1) (2 0 1 0) (2 1 0 0)			$k = 34$	(0 0 0 1) (3 0 0 0)

Table 6.4. The results of running Version 2 of Algorithm 5.1 with three sets of Dörfler marking parameters for the model problem in Experiment 3.

two other cases.

If we now compare numerical results in cases (ii) and (iii) (i.e., for fixed $\theta_X = 0.5$ and varying $\theta_{\mathfrak{P}}$), we can see little difference in the spatial and parametric approximations generated by both versions of the algorithm in these two cases (see the last two columns in Tables 6.3 and 6.4). This suggests that for this larger value of θ_X, both versions do not perform optimally, irrespective of the value of $\theta_{\mathfrak{P}}$ (which is consistent with our observations in Experiments 1 and 2). Furthermore, Version 1 reaches the tolerance faster than Version 2 in case (ii) and in case (iii). This conclusion is in agreement with numerical results for the parametric problem with spatially singular solution in Experiment 2, and it indicates that in terms of efficiency, Version 2 is more sensitive to over-refined triangulations than Version 1.

By looking at Figure 6.11 we observe that in case (i) (i.e., for $\theta_X = 0.2$), both versions of the algorithm converge faster during mesh refinement steps than in cases (ii) and (iii) (where $\theta_X = 0.5$). When running Version 2 in case (i), the convergence rate during mesh refinement steps is very close to the optimal one of $\mathcal{O}(N^{-1/2})$. On the other hand, the overall decay rate for total error estimates is about $\mathcal{O}(N^{-0.4})$ in all three cases (i)–(iii) and for both versions of the algorithm.

Finally, focusing on cases (i) and (ii), we compute the effectivity indices θ_k via (6.3). Similarly to the procedure used for Experiment 2, we employ a reference Galerkin solution computed with quadratic (P_2) approximations over a fine grid and using the index set generated by Version 2 in case (i). The resulting effectivity indices are plotted in Fig-

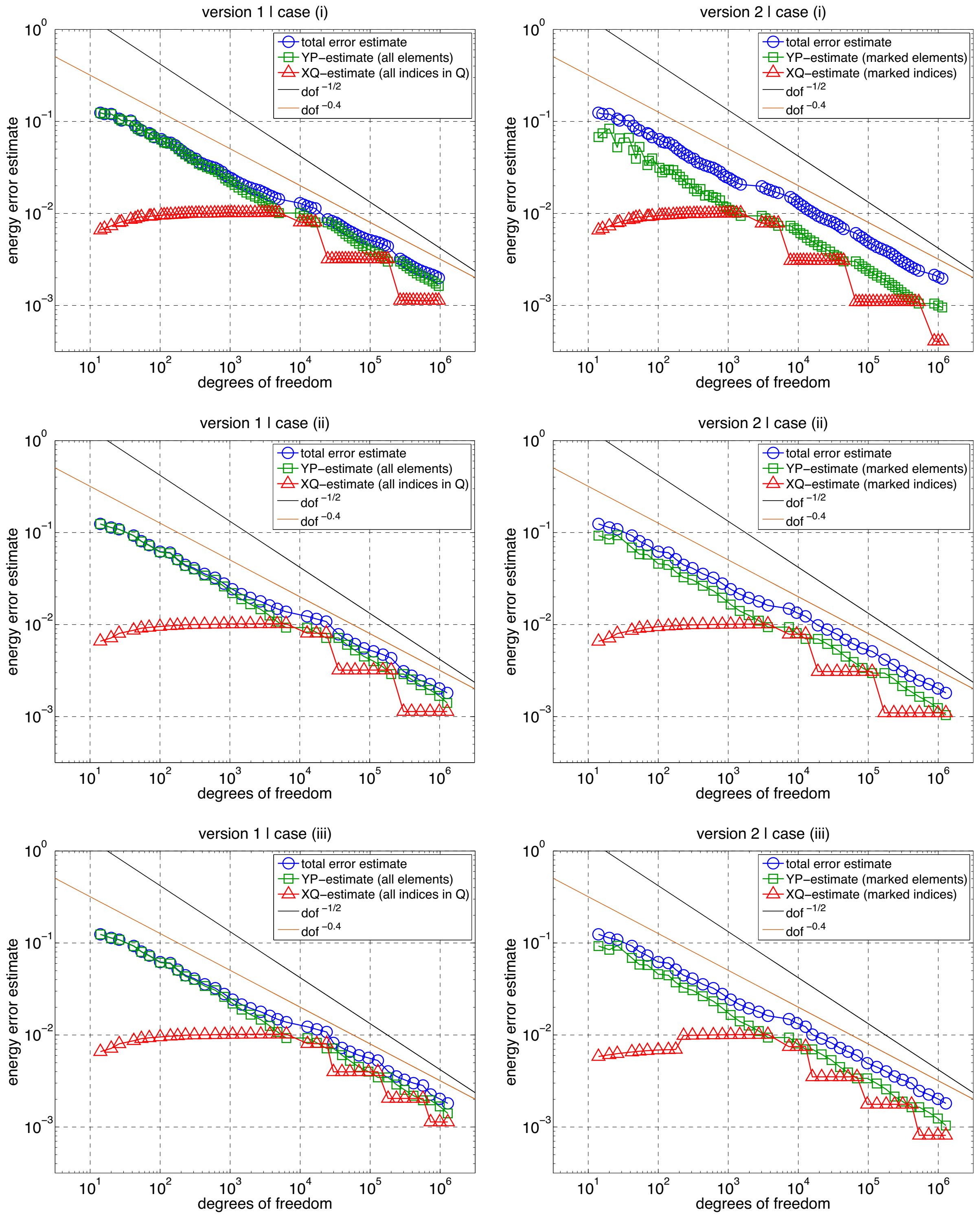


Figure 6.11. Energy error estimates at each step of the adaptive algorithm for the model problem in Experiment 3.

ure 6.12 (case (i)) and Figure 6.13 (case (ii)). As in previous experiments, the effectivity indices are less than unity for all iterations; for this model problem, however, they tend to increase as iterations progress and reach the values close to 0.9.

The results of Experiment 3 lead us to the same conclusion about the efficiency of Version 2 of Algorithm 5.1 as did the results of Experiments 1 and 2: provided that a sufficiently small (*spatial*) marking threshold θ_X is selected ($\theta_X = 0.2$ in all three experi-

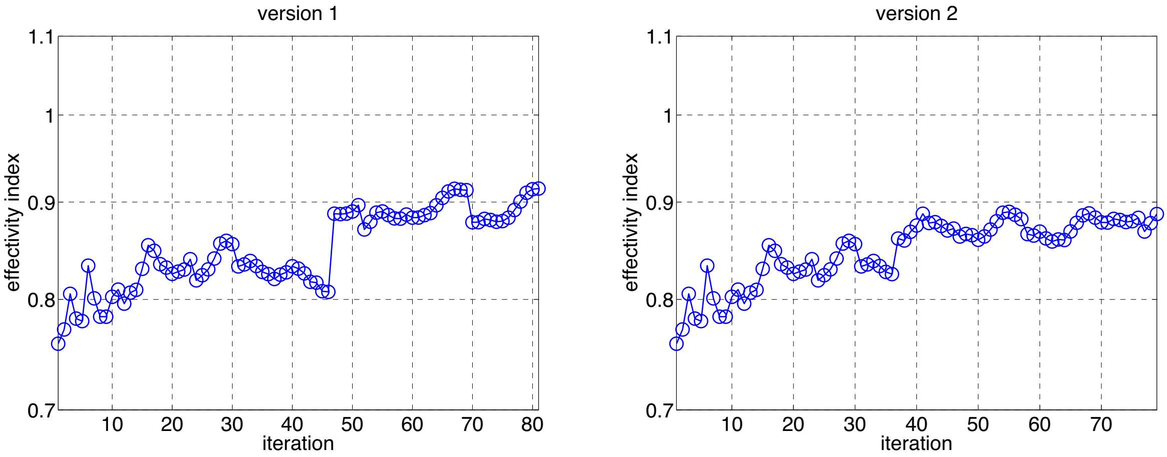


Figure 6.12. The effectivity indices for the sGFEM solutions in Experiment 3 at each iteration of the adaptive algorithm with $\theta_X = 0.2$, $\theta_{\mathfrak{P}} = 0.9$ (case (i)).

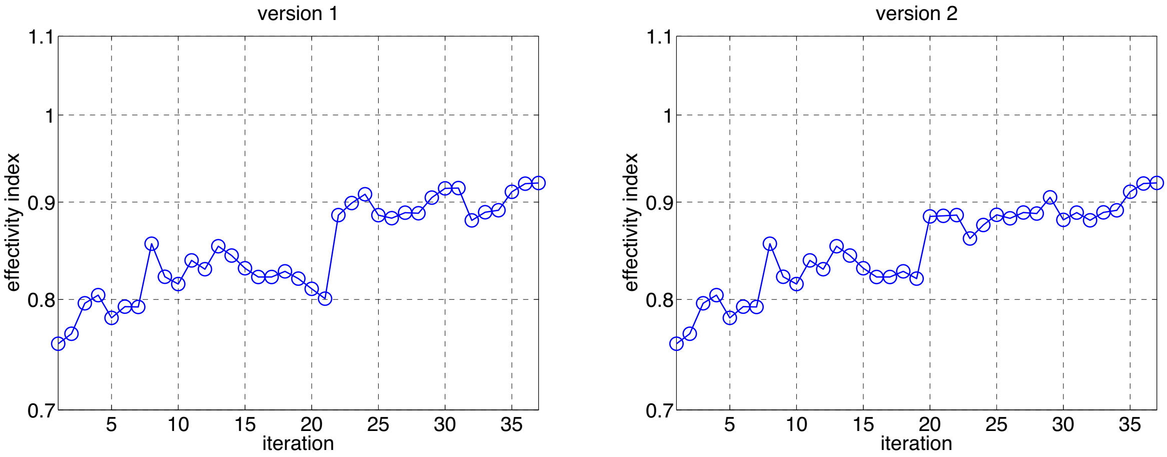


Figure 6.13. The effectivity indices for the sGFEM solutions in Experiment 3 at each iteration of the adaptive algorithm with $\theta_X = 0.5$, $\theta_{\mathfrak{P}} = 0.9$ (case (ii)).

ments), Version 2 reaches the set tolerance faster and leads to the error estimates decaying with the optimal rate during mesh refinement steps. Interestingly, this conclusion holds for parametric model problems with spatially singular solutions as well as for problems with spatially regular solutions. In addition, the results of Experiment 3 suggest that higher values of the (*parametric*) marking threshold $\theta_{\mathfrak{P}}$ should be preferred, as they lead to less number of iterations and therefore faster computations (this conclusion holds for both versions of the algorithm). We note, however, that selecting $\theta_{\mathfrak{P}} = 1$ (i.e., marking *all* indices in the detail index set $\mathfrak{Q}$) is generally not desirable, because the estimates $\|e^{(\mu)}_{X\mathfrak{Q}}\|_{B_0}$ ($\mu \in \mathfrak{Q}$) vary significantly in their magnitude, with a high proportion of them being much smaller than a few large estimates. Therefore, marking the indices in $\mathfrak{Q}$ that correspond to very small estimates and subsequently adding these indices to the index set $\mathfrak{P}$ does not lead to significant error reduction but increases computational cost. Finally, and this is again in agreement with results of the previous experiments, the overall convergence rate is essentially the same for both versions of the algorithm and for all considered cases of marking thresholds; note that this rate is slightly higher in the case of the model problem in Experiment 3.

7 Concluding remarks

The development of efficient adaptive algorithms is critical for effective numerical solution of elliptic PDEs with correlated random data. An important contribution of this paper is that it presents an innovative adaptive algorithm driven by precise estimates of the error reductions that would be achieved by pursuing different refinement strategies. There are two distinctive features in our approach. Firstly, the approximation error is controlled in the algorithm via hierarchical a posteriori error estimates that are shown to be reliable, efficient, and computationally effective. Secondly, the error reduction estimates are used in the algorithm (specifically, in its second version) not only to guide adaptive refinement but also to choose between spatial and parametric refinement at each iteration of the algorithm. As demonstrated in extensive numerical experiments for parametric PDE problems, this latter feature ensures well balanced refinement of spatial and parametric components of Galerkin approximations and leads to optimal convergence during mesh refinement steps, provided that the marking thresholds θ_X, $\theta_{\mathfrak{P}}$ are selected appropriately (the experiments suggest to choose $\theta_X = 0.2$ and $\theta_{\mathfrak{P}} \in \{0.8,\ 0.9\}$ for spatially regular as well as for spatial singular problems). Finally, the software that implements our adaptive algorithm for a range of parametric elliptic PDEs is available online and can be used to reproduce numerical results presented in the paper.

References

[1] M. Ainsworth and J. T. Oden, *A posteriori error estimation in finite element analysis*, Pure and Applied Mathematics (New York), Wiley, 2000.

[2] I. Babuška, R. Tempone, and G. E. Zouraris, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM J. Numer. Anal., 42 (2004), pp. 800–825.

[3] R. E. Bank and A. Weiser, *Some a posteriori error estimators for elliptic partial differential equations*, Math. Comp., 44 (1985).

[4] E. Bänsch, *Local mesh refinement in* 2 *and* 3 *dimensions*, Impact Comput. Sci. Engrg., 3 (1991), pp. 181–191.

[5] A. Bespalov, C. E. Powell, and D. Silvester, *Energy norm a posteriori error estimation for parametric operator equations*, SIAM J. Sci. Comput., 36 (2014), pp. A339–A363.

[6] A. Bespalov and L. Rocchi, *Stochastic T-IFISS*, January 2018. Available online at `http://web.mat.bham.ac.uk/A.Bespalov/software/index.html#stoch_tifiss`.

[7] A. Bespalov and D. Silvester, *Efficient adaptive stochastic Galerkin methods for parametric operator equations*, SIAM J. Sci. Comput., 38 (2016), pp. A2118–A2140.

[8] P. Binev, W. Dahmen, and R. DeVore, *Adaptive finite element methods with convergence rates*, Numer. Math., 97 (2004), pp. 219–268.

[9] W. Dörfler, *A convergent adaptive algorithm for Poisson's equation*, SIAM J. Numer. Anal., 33 (1996), pp. 1106–1124.

[10] M. Eigel, C. J. Gittelson, C. Schwab, and E. Zander, *Adaptive stochastic Galerkin FEM*, Comput. Methods Appl. Mech. Engrg., 270 (2014), pp. 247–269.

[11] ——, *A convergent adaptive stochastic Galerkin finite element method with quasi-optimal spatial meshes*, ESAIM Math. Model. Numer. Anal., 49 (2015), pp. 1367–1398.

[12] M. Eigel and C. Merdon, *Local equilibration error estimators for guaranteed error control in adaptive stochastic higher-order Galerkin finite element methods*, SIAM/ASA J. Uncertain. Quantif., 4 (2016), pp. 1372–1397.

[13] M. Eigel, M. Pfeffer, and R. Schneider, *Adaptive stochastic Galerkin FEM with hierarchical tensor representations*, Numer. Math., 136 (2017), pp. 765–803.

[14] V. Eijkhout and P. Vassilevski, *The role of the strengthened Cauchy-Buniakowskiĭ-Schwarz inequality in multilevel methods*, SIAM Rev., 33 (1991), pp. 405–419.

[15] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, second ed., 2014.

[16] W. Gautschi, *Orthogonal polynomials: computation and approximation*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2004.

[17] R. G. Ghanem and P. D. Spanos, *Stochastic finite elements: a spectral approach*, Springer-Verlag, New York, 1991.

[18] I. Kossaczký, *A recursive approach to local mesh refinement in two and three dimensions*, J. Comput. Appl. Math., 55 (1994), pp. 275–288.

[19] G. J. Lord, C. E. Powell, and T. Shardlow, *An introduction to computational stochastic PDEs*, Cambridge Texts in Applied Mathematics, Cambridge University Press, New York, 2014.

[20] W. F. Mitchell, *Unified multilevel adaptive finite element methods for elliptic problems*, PhD thesis, Department of Computer Science, University of Illinois, 1988.

[21] M. C. Rivara, *Mesh refinement processes based on the generalized bisection of simplices*, SIAM J. Numer. Anal., 21 (1984), pp. 604–613.

[22] C. Schwab and C. J. Gittelson, *Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs*, Acta Numer., 20 (2011), pp. 291–467.

[23] K. G. Siebert, *Mathematically founded design of adaptive finite element software*, in Multiscale and adaptivity: modeling, numerics and applications, vol. 2040 of Lecture Notes in Math., Springer, Heidelberg, 2012, pp. 227–309.

[24] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Rrefinement Techniques*, Adv. Numer. Math., Wiley-Teubner, 1996.

[25] R. Verfürth, *A posteriori error estimation techniques for finite element methods*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2013.