ADAPTIVE ALGORITHMS FOR PARTIAL DIFFERENTIAL EQUATIONS WITH PARAMETRIC UNCERTAINTY

by

LEONARDO ROCCHI

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

February 2019

Supervisor: Dr Alex Bespalov

School of Mathematics
College of Engineering and Physical Sciences
The University of Birmingham

To my family,
for believing in me more than I was able of

Abstract

In this thesis, we focus on the design of efficient adaptive algorithms for the numerical approximation of solutions to elliptic partial differential equations (PDEs) with parametric inputs. Numerical discretisations are obtained using the stochastic Galerkin Finite Element Method (SGFEM) which generates approximations of the solution in tensor product spaces of finite element spaces and finite-dimensional spaces of multivariate polynomials in the random parameters.

Firstly, we propose an adaptive SGFEM algorithm which employs reliable and efficient *hierarchical* a posteriori energy error estimates of the solution to parametric PDEs. The main novelty of the algorithm is that a balance between spatial and parametric approximations is ensured by choosing the enhancement associated with dominant error reduction estimates.

Next, we introduce a *two-level* a posteriori estimate of the energy error in SGFEM approximations. We prove that this error estimate is reliable and efficient. Then we provide a rigorous convergence analysis of the adaptive algorithm driven by two-level error estimates. Four different marking strategies for refinement of stochastic Galerkin approximations are proposed and, in particular, for two of them, we prove that the sequence of energy errors computed by associated algorithms converges linearly.

Finally, we use duality techniques for the goal-oriented error estimation in approximating linear quantities of interest derived from solutions to parametric PDEs. Adaptive enhancements in the proposed algorithm are guided by an innovative strategy that combines the error reduction estimates computed for spatial and parametric components of corresponding primal and dual solutions.

The performance of all adaptive algorithms and the effectiveness of the error estimation strategies are illustrated by numerical experiments. The software used for all experiments in this work is available online.

Acknowledgements

First of all, I would like to thank my supervisor, Alex Bespalov, for his continuous guidance, patience, and support during the last three years and half. I am very grateful to him for being always available and for his helpful advices about my research as well as, among other things, for the methodical attention to details that he taught me.

I also would like to express my gratitude to Dirk Praetorius and Michele Ruggeri for their valuable contribution to some of the theoretical results reported in this thesis.

Of course, I have to give thanks to my parents, for their encouragement and unconditional support during my doctoral studies. I also would like to thank my brother Jacopo. I will not forget how important has been his presence during my first year in Birmingham.

Finally, a special thanks goes to Teresa. She has been my brightest light during the dark moments.

Contents

1 Introduction **1**

1.1 Topics of the thesis 5

1.2 Main contributions of the thesis 7

1.3 Outline of the thesis 9

2 Preliminaries **11**

2.1 Function spaces 11

2.1.1 Sobolev spaces 12

2.1.2 Lebesgue-Bochner spaces 13

2.1.3 Tensor products of Hilbert spaces 13

2.2 Discrete function spaces 14

2.2.1 Triangulations 15

2.2.2 Piecewise polynomial spaces 16

2.3 Adaptive mesh-refinement in the finite element setting 17

2.3.1 Marking strategies 18

2.3.2 Newest vertex bisection 19

3 Random fields **23**

3.1 Random variables and probability spaces 23

3.2 Definition of random field 25

3.3 Representation of random fields 26

3.3.1 Karhunen-Loève expansions 26

3.3.2 Polynomial Chaos expansions 29

4 Discretisation of elliptic problems with parametric uncertainty **31**

4.1 Parametrisation of random inputs 32

4.1.1 The abstract setting of parametric operator equations 32

4.1.2 Parametric model problem 33

4.1.3 Main assumptions . . . 34
4.1.4 Weak formulation . . . 35
4.2 Stochastic Galerkin Finite Element Method . . . 37
4.2.1 Orthogonal polynomials in the parameter space . . . 38
4.2.2 Discrete weak formulation . . . 42
4.2.3 Stochastic Galerkin linear system . . . 44

5 Adaptive algorithms driven by hierarchical a posteriori error estimates 49
5.1 Enrichments via hierarchical basis . . . 50
5.1.1 Enriched tensor product spaces . . . 50
5.1.2 Enhanced Galerkin solutions . . . 52
5.1.3 Saturation assumption . . . 53
5.2 Hierarchical error estimate . . . 53
5.2.1 Error estimation using enriching subspaces . . . 54
5.2.2 Estimates of the error reduction . . . 56
5.2.3 Suitable detail index sets . . . 58
5.3 Adaptive SGFEM algorithm . . . 59
5.3.1 Computation of spatial and parametric estimates . . . 60
5.3.2 Marking strategy and refinements . . . 64
5.3.3 Adaptive loop . . . 64
5.4 Numerical experiments . . . 67
5.4.1 Setup of the experiments . . . 68
5.4.2 Experiment 1 - Spatially regular solution on square domain . . . 69
5.4.3 Experiment 2 - Spatially singular solution on L-shaped domain . . . 74
5.4.4 Experiment 3 - Spatially singular solution on slit domain . . . 78

6 Adaptive algorithms driven by two-level a posteriori error estimates 85
6.1 Two-level error estimate . . . 86
6.1.1 Main result . . . 87
6.1.2 Auxiliary lemmas . . . 89
6.1.3 Proof of the efficiency and reliability of the two-level estimate . . . 94
6.2 Adaptive SGFEM algorithms . . . 96

6.2.1 Local error contributions . . . 96
6.2.2 Schematic adaptive loop . . . 97
6.2.3 Estimates of the error reduction . . . 98
6.2.4 Marking criteria . . . 99
6.3 Analysis of convergence of the adaptive algorithm . . . 102
6.3.1 Convergence results . . . 103
6.3.2 Linear convergence of the energy errors . . . 104
6.4 Numerical experiments . . . 106
6.4.1 Experiment 1 - Comparison with hierarchical estimates . . . 106
6.4.2 Experiment 2 - Comparison of computational costs . . . 109

7 Adaptive algorithms for goal-oriented error estimation 113
7.1 Goal-oriented a posteriori error estimation . . . 114
7.1.1 Abstract setting . . . 114
7.1.2 Extension to the parametric setting . . . 115
7.2 A goal-oriented adaptive algorithm . . . 116
7.2.1 Local error estimates in the energy norm . . . 117
7.2.2 Marking strategy . . . 117
7.2.3 Error reduction in the product of energy norms . . . 120
7.2.4 Goal-oriented adaptive loop . . . 122
7.3 Numerical experiments . . . 123
7.3.1 Setup of the experiments . . . 124
7.3.2 Experiment 1 - Estimation of directional derivatives on square domain . . . 125
7.3.3 Experiment 2 - Estimation of directional derivatives on L-shaped domain . . . 130
7.3.4 Experiment 3 - Pointwise estimation on slit domain . . . 135

8 Concluding remarks 141

Appendix A Numerical experiment of Section 6.4.2 (extended version) 145

Appendix B Stochastic T-IFISS package 149

List of references 159

Notation

$d\mathbf{x}$	Lebesgue measure;
a.e.	almost everywhere (with respect to the Lebesgue measure);
$\mathbb{R}^d$	Euclidean d-dimensional space;
$\mathbb{N}$	set of natural numbers excluding zero;
$\mathbb{N}_0$	set of natural numbers including zero;
$\#$	cardinality;
δ_{ij}	Kronecker symbol;
∂D	boundary of a domain $D \subset \mathbb{R}^d$;
χ_D	characteristic function of D;
D^α	differential operator of order $\lvert\alpha\rvert$;
∇	gradient operator;
$\lVert\cdot\rVert_X$	norm of a Banach space X;
$\mathcal{H}'$	dual of a Hilbert space $\mathcal{H}$;
$(\cdot,\cdot)$	inner product in a Hilbert space $\mathcal{H}$;
$\langle\cdot,\cdot\rangle$	duality pairing between $\mathcal{H}'$ and $\mathcal{H}$;
$C^k(D)$	space of functions on D with continuous derivatives up to k;
$C_0^k(D)$	subspace of $C^k(D)$ of functions with compact support;
$\mathcal{L}(X,Y)$	space of bounded linear operators from X to Y;
Ω	sample space;
$\mathcal{F}(\Omega)$	σ-algebra of the events;
$\mathbb{P}$	probability measure;
$(\Omega,\mathcal{F}(\Omega),\mathbb{P})$	probability space;
$\mathbb{P}_Y$	probability distribution of a random variable Y on Ω;
ρ_Y	probability density function of a random variable Y on Ω;
i.i.d.	independent and identically distributed;
Γ	parameter space;
$\mathcal{B}(\Gamma)$	Borel σ-algebra over Γ;
$L^2_\pi(\Gamma)$	Lebesgue space of square integrable π-measurable functions over Γ;
$L^2_\pi(\Gamma;X)$	Lebesgue-Bochner space of π-strongly measurable functions from Γ to X;

$H^1(D)$ Sobolev space of L^2-functions with square integrable first-order weak derivatives;

$H^1_0(D)$ subspace of $H^1(D)$ of functions vanishing on ∂D in the sense of traces;

$\mathcal{T}$ triangulation of a bounded Lipschitz polygonal domain $D \subset \mathbb{R}^2$;

$\mathcal{N}(\mathcal{T})$, $\mathcal{N}^\circ(\mathcal{T})$ set of total and interior vertices of $\mathcal{T}$, respectively;

$\mathcal{E}(\mathcal{T})$, $\mathcal{E}^\circ(\mathcal{T})$ set of total and interior edges of $\mathcal{T}$, respectively;

$\mathcal{N}^+$ set of midpoints of interior edges of $\mathcal{T}$;

$\mathcal{P}_k$ space of polynomials of total degree less than or equal to k in d variables;

$\mathcal{S}^k(\mathcal{T})$ space of globally continuous piecewise polynomials in $\mathcal{P}_k$ over each element of $\mathcal{T}$;

$\mathcal{S}^k_0(\mathcal{T})$ subspace of functions in $\mathcal{S}^k(\mathcal{T})$ which vanish on the boundary ∂D;

X first-order finite element space $\mathcal{S}^1_0(\mathcal{T})$;

Y first-order detail finite element space over $\mathcal{T}$;

$\mathcal{I}$ subset of $\mathbb{N}_0^{\mathbb{N}}$ of finitely supported indices;

$\varepsilon^{(m)}$ Kronecker delta index;

$\mathcal{P}_\mathcal{P}$ finite-dimensional polynomial space associated with a finite index set $\mathcal{P} \subset \mathcal{I}$;

$N_\mathcal{P}$ cardinality of a finite index set $\mathcal{P}$;

$M_\mathcal{P}$ number of active parameters of a finite index set $\mathcal{P}$;

$\mathcal{Q}$ detail index set;

$\oplus$ direct sum of spaces;

$\otimes$ tensor product of Hilbert spaces or Kronecker product of matrices;

$\eta_{X\mathcal{P}}$ hierarchical a posteriori error estimate;

$\tau_{X\mathcal{P}}$ two-level a posteriori error estimate;

$\theta_X, \theta_\mathcal{P}$ spatial and parametric threshold parameters of a marking strategy, respectively;

$\mathcal{M}$ set of marked elements, midpoints, or edges;

$\mathfrak{M}$ subset of $\mathcal{Q}$ of marked indices.

Chapter 1

Introduction

Nowadays, mathematical models and computer simulations of problems depending on uncertain parameters are indispensable in science and many engineering applications. Partial differential equations (PDEs) with parametric uncertainty are ubiquitous in such mathematical models as they evermore represent the starting point of the investigation and play a major role in the modelling of physical phenomena. In these PDEs, diffusion coefficients, source terms, initial and boundary conditions are model input data that can be affected by uncertainty. In principle, this uncertainty can be simply due to the incomplete knowledge of the quantities in the model. This is the case of uncertainties typically referred to as being *epistemic*. For instance, in modelling groundwater flow in porous medium, subsurface quantities such as the permeability and porosity are not really random in nature: they are inaccessible or may not be known everywhere in the given domain. On the other hand, there are situations in which the uncertainty may come from the intrinsic variability in the physical system, as, for example, mechanical properties in linear elasticity problems or the action of the wind in geophysics models. These are the cases of the so-called *aleatoric* uncertainties. The need for reliable uncertainty quantification is therefore essential and typically achieved by supplying mathematical models with a probabilistic framework.

In the case of PDE problems whose physical input quantities vary in space, the modelling is usually performed by means of random fields (see [88]). The numerical computation of solutions to such PDEs involves three main stages. First, the input random field needs to be appropriately represented in order to numerically handle the uncertainty of the problem. Typically, the random field is parametrised by a large (possible infinite) number of random variables. For example, this can be achieved by using Karhunen-Loève (KL) expansions (see, e.g., [87, 88]), when the dependence on the parameters is linear, and generalised polynomial chaos (gPC) expansions (see, e.g., [133, 134, 135]) which allow more general dependence on the parameters. Second, an efficient

numerical method has to be utilised to discretise the parametric PDE and a robust solver has to be employed to solve the system arising from the discretisation. The third stage involves the estimation of the error due to the discretisation, or in other words, the assessment of the accuracy of the computed approximation.

Well-established numerical methods to solve parametric PDEs can be divided into two general broad classes: *non-intrusive* and *intrusive* methods (see [79]). Non-intrusive methods are typically suitable for parallelisation of deterministic solvers that can be used as building blocks of legacy codes. Most representative methods of this class are *Monte Carlo* (MC) methods (see, e.g., [38, 114]) in which the desired stochastic moment of the solution to the PDE is derived by averaging the spatial approximations obtained by sampling independent and identically distributed (i.i.d.) realisations of the input random data according to the assumed statistics. On the one hand, using MC methods requires minimal assumptions for the well-posedness of the parametric problem and their implementation is elementary based on solving independent deterministic problems. On the other hand, MC methods can result in expensive computations and, most importantly, yield approximations that converge slowly. For this reason, other popular sampling-based methods which generally improve the performance of classic MC approach, have attracted a lot of attention in recent years. These are, e.g., *quasi*-Monte Carlo (QMC) methods (see, e.g., [38, 51]), *multi-level* Monte Carlo (MLMC) methods (see, e.g., [18, 43, 126, 73]), *multi-index* Monte Carlo (MIMC) methods (see, e.g., [80]), and *stochastic collocation* (SC) methods (see, e.g., [5, 102, 103, 101]).

Unlike non-intrusive methods, intrusive methods require partial redesign of the algorithms as existing deterministic codes are modified in order to couple together both spatial and stochastic degrees of freedom at early stage of the code. Most popular intrusive methods are stochastic Galerkin methods which date back to the pioneering work [72]. These methods are based on a variational (weak) formulation posed on an appropriate Lebesgue-Bochner space for the given stochastic problem. In particular, a gPC expansion of the solution, which takes into account the stochastic approximation, is combined together with a Galerkin projection onto a finite-dimensional space for spatial approximations. In other words, the coefficients of the solution are approximated with respect to a basis of functions (i.e., polynomials) depending on the parameters. This represents a major difference to sampling-based methods such as MC, QMC, and MLMC methods: while these methods are used to target directly stochastic moments of the unknown solution, the gPC expansion arising from applying stochastic Galerkin methods also provides an *explicit* parametric

representation thereof.

Under the finite element method (FEM) setting (see [42, 34, 35, 125]), a very efficient alternative to sampling-based methods is represented by the *stochastic Galerkin finite element method* (SGFEM) (see, e.g., [48, 49, 8, 9]). This is a powerful tool which received a considerable attention over the last two decades. As part of intrusive stochastic Galerkin methods, in the SGFEM, numerical approximations are sought in tensor product spaces of finite element spaces associated with the physical domain and spaces of (multivariate) polynomials over a finite-dimensional manifold in the *parameter domain*. For the class of PDE problems whose data depend linearly on random parameters, the SGFEM has been shown, on the one hand, to be immune to the 'curse of dimensionality' (see, e.g., [44, 45]) and, on the other hand, to outperform standard classic sampling-based methods. For example, so-called multi-level (or sparse tensor) versions of SGFEM discretisations converge independently of the dimension of parameter spaces, thus achieving convergence rates which are superior to those of MC approximations (see [30, 118]); in particular, right implementation of the SGFEM leads to optimal rates as if the stochastic problem was parameter-free (see, e.g., [47]).

A fundamental aspect of any simulation is the cost associated with running the chosen numerical method. For parametric PDEs, regardless of growing computational power of modern computers which allow the treatment of high-dimensional problems, if a large number of random variables is used to represent the input data of the parametric problem, and highly refined spatial grids are used for spatial approximations on the physical domain, then computing the solution (or stochastic moments) may become prohibitively expensive. This is true, in particular, for both sampling-based and stochastic Galerkin methods. In this respect, however, much work has been done in recent years. For example, it is true that the cost associated with using the SGFEM is high. This is because the SGFEM allows the simultaneous discretisation of both physical and parametric spaces, and thus result in huge linear systems normally many orders of magnitude larger than deterministic subproblems arising from, e.g., MC methods. Nonetheless, it is also true that the resulting matrix arising from applying the SGFEM is highly block-sparse (see, e.g., [65]). Firstly, this means that, effectively, such matrix does not need to be assembled. Secondly, according to its symmetry and definiteness, the linear system is suitable to be solved by iterative solvers (e.g., CG, MINRES, and GMRES, see [128, 129, 122]) performing efficient matrix-vector operations of single blocks. Further to that, to speed up the computation, iterative solvers are

always coupled with preconditioning strategies (see, e.g., [106, 107, 127]).

In computational PDEs, especially under the finite element framework, the design and theoretical analysis of adaptive FEM algorithms have received a significant attention for deterministic (see, e.g., [52, 96, 97, 31, 93, 98, 40]), and more recently, for parametric PDE problems. In particular, along with the computation of numerical solutions, a major role in this area is played by the *a posteriori* error estimation (see, e.g., [2, 131]). It is well known in the finite element community that adaptive strategies based on rigorous a posteriori error analysis of computed solutions provide an effective mechanism for building approximation spaces and accelerating convergence. In particular, numerical adaptive algorithms should be designed so as to identify a finite set of most important parameters to be incorporated into the basis of the approximation space and, at the same time, spatial and stochastic components of approximations should be judiciously chosen and incrementally refined in the course of numerical computation. This is desirable in order to compute approximations of the solution (or other quantities of interest different from the solution) up to a prescribed accuracy (engineering tolerance) with minimal computational work. However, this presents a number of theoretical and practical challenges. In fact, a posteriori error estimation techniques rely heavily on how the approximation error is estimated and controlled. In this respect, most common techniques primarily focus on the estimation of the error in a suitable norm, typically, the (energy) norm induced by the bilinear form of the variational formulation associated with the PDE.

Within the SGFEM setting for PDE problems with parametric inputs, several adaptive strategies, based on the estimation of global energy norm of the errors, are used to enhance the computed solution and drive the convergence of approximations. Such strategies are developed by extending the a posteriori error estimation techniques commonly used for deterministic problems (see, e.g., [50, 16, 2, 96, 97]) to the parametric setting. For example, explicit residual-based a posteriori error estimates provide spatial and stochastic error indicators for adaptive refinement in [77, 54, 55]; implicit error estimators are used in [132] for the SGFEM based on multi-element gPC expansions; local equilibration error estimates are utilised in [56]; hierarchical error estimates and associated estimates of error reduction drive adaptive algorithms proposed in [29, 27, 47]. It is worth mentioning that in contrast to the design, the convergence analysis of algorithms for parametric PDEs is, however, much less developed. Among the most significant contributions, the convergence of an adaptive SGFEM algorithm driven by residual-based estimates is proved

in [55] under additional assumptions about enforcing spatial refinements also during iterations where enrichment of parametric discretisation is performed; moreover, the quasi-optimality of the generated sequence of grids (in a suitable sense) is established.

In many other practical applications, however, one is not interested in globally approximating the solution and estimating the associated error in some suitable norm. The estimation of the error in computing some specific feature of the solution may be, indeed, more useful to the application. For instance, simulations may target a feature of the solution localised on some part of the computational domain, e.g., the approximation of pointwise values or the approximation of stochastic moments on a region where the solution exhibits spatial singularities. In this case, the global energy norm of the error does not provide any meaningful information. Error estimation strategies thus need to address the approximation of a prescribed quantity of interest (or goal quantity) which can be typically represented by some linear functional of the solution. In the deterministic setting, such *goal-oriented* techniques are well-established (see, e.g., [21, 109, 22, 74, 11]) whereas relatively little work has been done in the parametric setting; notice that in this latter case, the quantity of interest is parametric since it depends on the parameters through the solution itself. In the framework of non-intrusive methods, goal-oriented error estimation techniques and associated adaptive algorithms are proposed in [57] for the MLMC method and in [3] for the SC sampling, where, in particular, the authors proposed a procedure to estimate the quantity of interest at each collocation point. Under the SGFEM setting, error estimation of linear functionals of solutions is addressed in [90] and, for nonlinear problems, in [37]; these works naturally extend deterministic *dual-weighted* residual methods to parametric PDEs. In addition, goal-oriented estimates derived from generic surrogate approximations (either intrusive or non-intrusive) are introduced in [36].

1.1 Topics of the thesis

In this thesis, we mainly deal with the design of efficient adaptive strategies for the numerical discretisation of parametric PDEs. The model problem under consideration is a parametric steady-state diffusion equation with homogeneous Dirichlet boundary conditions on a spatial two-dimensional polygonal domain. We further assume that the source term is deterministic rather than being parametric. We consider the case of domains on which the solution may be

either regular (as for simple square domains) or may exhibit a corner singularity (as in case of L-shaped or slit domains). The input diffusion coefficient is represented by a spatially varying random field depending linearly on an infinite countable set of (random) parameters. These are defined as the images, in the parameter space, of infinite i.i.d. random variables on an underlying probability space. Note that our model problem is the same parametric equation considered in, e.g., [49, 9, 29, 54, 56], to name but a few.

We will recall the probability framework lying in the background of our investigation and how to handle numerically the input parametric coefficients by means of KL or gPC expansions of random fields. For the discretisation of the problem, the SGFEM is the numerical method that we consider throughout. The starting point is the variational (or weak) formulation that is derived from the parametric problem via direct application of the SGFEM. The main focus will be on the a posteriori error estimation, for computed SGFEM solutions, in the global energy norm and in estimating prescribed quantities of interest.

Our goal is the development, and in part the analysis, of adaptive SGFEM algorithms derived under the stardard SOLVE, ESTIMATE, MARK, REFINE paradigm of adaptive finite element methods (see, e.g., [104, 105]). Two kinds of a posteriori error estimates are used by such algorithms: a *hierarchical* error estimate, developed in [24, 29] and based on results from [16, 15, 12], and a novel *two-level* error estimate based on ideas from [100, 99, 62]. Hierarchical and two-level a posteriori estimates are primarily used for the estimation of the energy norm of the global error. We will show how they can be also used for the goal-oriented error estimation in the quantity of interest of SGFEM solutions. Both estimates are proved to be efficient and reliable for the estimation of the energy norm of the error under the so-called *saturation assumption* (see, e.g., [16, 12, 13, 39] for the deterministic setting). In addition, such estimates are used to identify dominant sources of discretisation error and guide the proposed adaptive algorithms by using various refinement strategies for the enhancement of either the spatial or the parametric component of the solution. For marking purposes, two popular strategies such as the maximum (see [10]) and the Dörfler (see [52]) marking strategies are considered; for spatial refinement of grids of the physical domain, we primarily focus on the newest vertex bisection (NVB) rule (see, e.g., [17, 84, 31, 124]).

The main tasks that we aim to pursue in this thesis are essentially three:

- designing an adaptive SGFEM algorithm which uses hierarchical a posteriori estimates for the energy error of computed solutions;

- designing an adaptive SGFEM algorithm which uses two-level a posteriori estimates for the energy error of computed solutions as well as investigating the convergence properties of such proposed algorithm;
- designing a goal-oriented adaptive algorithm for the error estimation of linear quantities of interest (different from the energy error) derived from computed solutions; the resulting total error estimate may be based on either hierarchical or two-level estimates.

For the above mentioned tasks, a large part of the work is dedicated to the results of extensive numerical experiments which aim at illustrating the computational aspects as well as the performance of the proposed adaptive algorithms. We emphasise that although we focus on a specific model problem, we try to keep the design of the algorithms as general as possible. In particular, they can be extended, with natural amendments, to more general PDE problems. These can be, for example, reaction-diffusion problems, PDEs with parametric source terms with affine dependence on the parameters, and, as in [82], parameter-dependent linear elasticity equations.

1.2 Main contributions of the thesis

The main contributions of the present work can be summarised as follows.

First, we propose an adaptive algorithm which employs hierarchical error estimates from [29] for the estimation of the energy error of computed SGFEM solutions to the parametric model problem. The enrichment of finite element spaces in the algorithm in [29] is based on uniform refinements of the spatial grid, hence allowing an efficient discretisation only of spatially regular problems. We extend this algorithm in order to deal with problems exhibiting spatial singularities. Furthermore, our proposed algorithm is designed so as to run in two different versions. A first, more traditional version, which is driven by *total* error estimates, and a second version, in which the Dörfler strategy firstly returns sets of marked spatial and parametric components of the two sources of discretisation error, and then the associated larger *error reduction* estimate indicates the type of enhancement. For both versions, numerical experiments show that the algorithm is efficient and able to ensure a balance between spatial and parametric approximations. Although adaptive algorithms for parametric PDEs already exist in the literature (see, e.g., [76, 55]), our algorithm is probably the first useful tool for such kind of problems under the framework of hierarchical a posteriori error estimation.

Second, a *two-level* estimate for the energy norm of the global error is introduced. The estimate is derived under the same hierarchical framework from [29] (see also [24]) and its construction is based on ideas from the deterministic setting (see [100, 99, 62]). We prove that such a novel estimate is both efficient and reliable. To ease the presentation of the analysis, our proof is based on the two-dimensional model problem under consideration, though it applies to any spatial dimension. Then we study the convergence of the associated adaptive SGFEM algorithm. In particular, four versions of the algorithm, based on four different marking strategies that combine both the maximum and the Dörfler strategies, are proposed. For all versions, by adopting the arguments from [98], it is proved that the computed sequence of two-level error estimates converges to zero (see [25]). We stress that such a result holds independently of the saturation assumption, and in particular, the analysis does not require extra assumptions on the refinement level of the underlying spatial grid as, for example, the assumption needed in [55]. Moreover, for two versions of the algorithm, in the spirit of [96], we prove that *linear* convergence of the energy errors is expected (yet assuming the saturation assumption in this case). These contributions fill a gap in the theoretical analysis of adaptive SGFEM algorithms for elliptic parametric PDEs.

Third, we use the ideas of deterministic goal-oriented error estimation (see [22, 74, 11]) and adaptivity to design and implement an efficient adaptive algorithm for approximating linear quantities of interest of the computed solution to our parametric model problem. In the algorithm, the SGFEM is used to approximate the solutions to both primal and dual problems and adaptive refinement is guided by an innovative strategy that combines the error reduction estimates associated with spatial and parametric components of the primal and dual solutions. Specifically, the marking is performed by employing and extending the strategy proposed in [67] to the parametric setting. Numerical experiments illustrate the performance and the effectiveness of such error estimation strategy.

Finally, we want to emphasise that all numerical aspects of the main contributions reported in this thesis, such as the implementation of the two-level error estimation and all proposed adaptive algorithms, have been developed in conjunction with the open source Matlab toolbox *Stochastic T-IFISS* [28] which is available online and has been used to run all numerical experiments. Of course, there exist many available pieces of software for general uncertain quantification, such as FERUM [33] and UQLab [89], as well as finite element packages that can be employed as a framework to solve PDEs problems with uncertain inputs (e.g., `p1afem` [69], ALBERTA [117], and

FEniCS [4]). Open source software packages implementing stochastic Galerkin methods, such as ALEA [58] and SGLib [136], are also freely available. In this respect, our Stochastic T-IFISS toolbox has been developed and designed to provide a computational laboratory for elliptic parametric PDEs. Besides supporting the investigation reported in this work, it represents a contribution to the set of existing packages for the study of parametric PDEs and can be used for academic research as well as teaching.

1.3 Outline of the thesis

In Chapter 2, we recall the definitions and properties of function spaces such as Sobolev spaces and Lebesgue-Bochner spaces (Section 2.1), and finite element spaces on conforming triangulations of bounded domains (Section 2.2). In addition, in Section 2.3, we describe two popular marking strategies and the NVB refinement rule adopted by all adaptive SGFEM algorithms presented in the thesis.

In Chapter 3, we introduce the probabilistic framework required to setup the study of PDEs with parametric inputs. This includes the definition of probability space, random variable, and random field (Sections 3.1 and 3.2) as well as how to represent the latter by means of Karhunen-Loève and polynomials chaos expansions (Section 3.3).

In Chapter 4, we present the elliptic parametric model problem that we considered in this work. We make all the required assumptions on the underlying probability space, the parameter space, and the input random field required for the well-posedness of the problem. Then we derive its weak formulation (Section 4.1). In Section 4.2, we introduce the approximation space and briefly describe the discretisation of the weak problem via the SGFEM and discuss some numerical implementation aspects.

Chapter 5 is dedicated to the design of adaptive SGFEM algorithms driven by hierarchical a posteriori error estimates. Initially, we recall the parametric framework from which such estimates are derived (Sections 5.1 and 5.2). Then, in Section 5.3, we present the adaptive SGFEM algorithm for the numerical solution to our parametric problem. We describe all modules composing the adaptive loop and present the two versions of the algorithm driven by dominant total error estimates and error reduction estimates, respectively. Section 5.4 presents the results of three illustrative numerical experiments posed on square, L-shaped, and slit domains.

In Chapter 6, the novel two-level error estimate is introduced and analysed. In Section 6.1, we set up the notation, define the estimate, and prove that it is both efficient and reliable for the estimation of the energy norm of the error. In Section 6.2, we present adaptive algorithms with four marking criteria, whereas in Section 6.3 we state the convergence of the computed sequence of two-level error estimates and prove the linear convergence of global energy errors. Numerical experiments are reported in Section 6.4, where we first compare the performance of adaptive algorithms driven by hierarchical and two-level estimates, and then compare the computational cost associated with employing the proposed marking strategies when using two-level estimates.

Chapter 7 deals with the design of an adaptive goal-oriented SGFEM algorithm for the numerical approximation of prescribed quantities of interests represented by linear functionals of the solution to the model problem. Firstly, we show how deterministic goal-oriented error estimation techniques can be easily applied to the parametric setting (Section 7.1). Then, a novel goal-oriented adaptive algorithm is presented in Section 7.2. The effectiveness of the error estimation strategy and the performance of the algorithm are demonstrated in Section 7.3 by three numerical experiments including the estimation of directional derivatives and approximated pointwise values of the solution.

Chapter 8 contains the concluding remarks about the present work.

In Appendix A we report the complete results of the numerical experiment in Section 6.4.2 about the comparison of computational cost of adaptive algorithms in Chapter 6.

Finally, Appendix B aims at highlighting and briefly describing the main components of the toolbox Stochastic T-IFISS which was used to perform all numerical experiments in the thesis.

Chapter 2

Preliminaries

In this chapter, we introduce some basic notation and recall many of the ingredients that will be used throughout the thesis. These include functions spaces, in particular, Sobolev spaces on bounded domains $D \subset \mathbb{R}^d$, $d \in \mathbb{N}$, as well as Lebesgue-Bochner spaces of functions taking values in arbitrary Banach spaces. Next, we recall the definition of conforming triangulations of a domain D and discrete polynomial (finite element) spaces defined on such triangulations. Finally, in the context of adaptive finite element methods for the numerical solution of partial differential equations, we describe two popular marking strategies for the selection of elements contributing with large errors to the approximation of the solution as well as a particular mesh-refinement technique, which is the one used by all adaptive algorithms proposed in this work.

2.1 Function spaces

Let (T, Σ, μ) be a measure space, where T is a non-empty set, Σ is a σ-algebra on T, and μ is a measure. For all $1 \leq p < \infty$, we denote by $L^p_\mu(T)$ the Lebesgue space of μ-measurable functions such that

$$\|v\|_{L^p_\mu(T)} := \left(\int_T |v(t)|^p \, d\mu \right)^{1/p} < +\infty. \tag{2.1}$$

For $p = \infty$, the L^p-norm is defined by $\|v\|_{L^\infty_\mu(T)} := \operatorname{ess\,sup}_{t \in T} |v(t)|$. Whenever $L^p_\mu(T)$ is a Hilbert space, $(\cdot,\cdot)_\mu$ denotes the inner product on $L^p_\mu(T)$ which induces the L^p-norm (2.1). For a domain $D \subset \mathbb{R}^d$, $d \in \mathbb{N}$, spatial points of D are denoted by $\mathsf{x} = (x_1, \ldots, x_d)$. Also, in this case, we denote Lebesgue spaces simply by $L^p(D)$, their norms by $\|\cdot\|_{L^p(D)}$ with the Lebesgue measure $d\mathsf{x}$ replacing $d\mu$ in (2.1), and for $p = 2$, we denote the associated inner product by $(\cdot,\cdot)_{L^2(D)}$. Furthermore, let

$\alpha := (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ be a multi-index of d non-negative integers. For a multi-index α,

$$D^\alpha := \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}, \tag{2.2}$$

denotes the differential operator of order $|\alpha| := \sum_{i=1}^d \alpha_i$. In the case $d = 2$, ∇ represents the gradient operator, i.e., the vector of first-order derivatives $\nabla := (D^{(1,0)}, D^{(0,1)})$ in x_1 and x_2.

2.1.1 Sobolev spaces

Let $C^k(D)$, $k \in \mathbb{N}_0 \cup \{\infty\}$, be the space of functions whose all derivatives D^α of orders $|\alpha| \le k$ are continuous on D. We say that a function $v \in L^p(D)$ has a *weak derivative* w of order $|\alpha|$ if

$$\int_D v(\mathbf{x}) D^\alpha \varphi(\mathbf{x}) d\mathbf{x} = (-1)^{|\alpha|} \int_D w(\mathbf{x}) \varphi(x) d\mathbf{x} \qquad \forall\, \varphi \in C_0^\infty(D), \tag{2.3}$$

where $C_0^\infty(D)$ denotes the space of functions in $C^\infty(D)$ with compact support in D. Note that if $v \in L^p(D)$ has continuous partial derivatives D^α in the classical sense (2.2), then D^α coincides with the weak derivative. However, a partial derivative may exist without existing in the classical sense. Hereafter, D^α will then refer to weak derivatives in general but also assumes the meaning of classical derivative as appropriate.

Sobolev spaces are defined as follows (see, e.g., [1, 35]).

Definition 2.1 (Sobolev space)**.** *Let $k \in \mathbb{N}_0$. The Sobolev space $W^{k,p}(D)$ is defined as*

$$W^{k,p}(D) := \left\{ v \in L^p(D) \;:\; \|v\|_{W^{k,p}} < \infty \right\},$$

where the Sobolev norm $\|\cdot\|_{W^{k,p}}$ is given by

$$\|v\|_{W^{k,p}} := \begin{cases} \left(\sum_{|\alpha| \le k} \|D^\alpha v\|_{L^p(D)}^p \right)^{1/p} & \text{if } 1 \le p < \infty, \\ \max_{|\alpha| \le k} \|D^\alpha v\|_{L^p(D)} & \text{if } p = \infty, \end{cases}$$

where D^α denotes weak derivatives (see (2.3)*).*

Clearly, $W^{0,p}(D) = L^p(D)$ for all $1 \le p \le \infty$. It is well known that, for $1 \le p \le \infty$, Sobolev spaces $W^{k,p}(D)$ are Banach spaces when equipped with the Sobolev norm $\|\cdot\|_{W^{k,p}(D)}$ (see, e.g., [1, Theorem 3.3]). We denote by $H^k(D)$ the space $W^{k,2}(D)$ which is a Hilbert space equipped with the inner product

$$(v, w)_{H^k(D)} := \sum_{|\alpha| \le k} (D^\alpha v, D^\alpha w)_{L^2(D)} \qquad \forall\, v, w \in H^k(D),\ \alpha \in \mathbb{N}_0^d.$$

Furthermore, for $k = 1$, we denote by $H_0^1(D)$ the closure of $C_0^\infty(D)$ in $H^1(D)$. In particular, if D has

a *Lipschitz* boundary ∂D (e.g., in the sense of [35, Definition 1.4.4]), then $H_0^1(D)$ can be identified with the space of functions $v \in H^1(D)$ such that $v|_{\partial D} = 0$, where $v|_{\partial D}$ has to be understood in the sense of *traces* (see, e.g., [35, Section 1.6]).

2.1.2 Lebesgue-Bochner spaces

Lebesgue-Bochner spaces are a generalisation of Lebesgue spaces to functions which take values in an arbitrary Banach space rather than in $\mathbb{R}$ (or $\mathbb{C}$); see, e.g., [115, 1]. These are the natural candidate spaces to consider when we look for solutions of PDEs depending on uncertain or parametric inputs (see Section 4.1.4).

Let (T, Σ, μ) be a measure space and X be a Banach space with norm $\|\cdot\|_X$. We say that a function $s : T \to X$ is *simple* if $s(t) = \sum_{k=1}^m \chi_{E_k}(t) v_k$, $t \in T$, where χ_{E_k} denotes the characteristic function of a μ-measurable subset E_k of T and $v_k \in T$ for $k = 1, \dots, m$, with $m \in \mathbb{N}$. Then, a function $v\colon T \to X$ is *strongly μ-measurable* if there exists a sequence $(s_n)_{n\in\mathbb{N}}$ of simple functions such that $s_n(t) \to v(t)$ as $n \to \infty$, for a.e. $t \in T$.

Definition 2.2 (Lebesgue-Bochner space)**.** *Let (T, Σ, μ) be a measure space and X be a Banach space with norm $\|\cdot\|_X$. The Lebesgue-Bochner space $L^p_\mu(T; X)$ is the Banach space defined as*

$$L^p_\mu(T;X) := \left\{ v\colon T \to X \ : \ v \text{ is } \mu\text{-strongly measurable and } \|v\|_{L^p_\mu(T;X)} < \infty \right\}, \tag{2.4}$$

where the Bochner norm $\|\cdot\|_{L^p_\mu(T;X)}$ is given by

$$\|v\|_{L^p_\mu(T;X)} := \begin{cases} \left(\int_T \|v(t)\|_X^p \, d\mu \right)^{1/p} & \text{if } 1 \le p < \infty, \\ \operatorname{ess\,sup}_{t\in T} \|v(t)\|_X & \text{if } p = \infty. \end{cases}$$

If $p = 2$ and X is a separable Hilbert space with inner product $(\cdot,\cdot)_X$, then $L^2_\mu(T;X)$ is itself a Hilbert space with inner product defined as $(u, v)_{L^2_\mu(T;X)} := \int_T (u(t), v(t))_X \, d\mu$ for all $u, v \in L^2_\mu(T;X)$ (see, e.g., [88, Proposition 1.34]).

2.1.3 Tensor products of Hilbert spaces

Throughout the thesis, we will often make use of tensor products of Hilbert spaces when defining the trial and test spaces for discrete weak formulations of parametric PDEs. Let us briefly recall the definition following the construction given in [111, Section II.4]; see also, e.g., [85, Chapter 1] and [115].

Let $\mathcal{H}_1$ (resp. $\mathcal{H}_2$) be a Hilbert space equipped with the inner product $(\cdot,\cdot)_{\mathcal{H}_1}$ (resp. $(\cdot,\cdot)_{\mathcal{H}_2}$). For $v_1 \in \mathcal{H}_1$ and $v_2 \in \mathcal{H}_2$, let $v_1 \otimes v_2$ be the bilinear form on $\mathcal{H}_1 \times \mathcal{H}_2$ defined by

$$(v_1 \otimes v_2)(\psi_1, \psi_2) := (\psi_1, v_1)_{\mathcal{H}_1}(\psi_2, v_2)_{\mathcal{H}_2} \qquad \forall\, \psi_1 \in \mathcal{H}_1,\ \forall\, \psi_2 \in \mathcal{H}_2.$$

Now, consider the space $\mathcal{C}$ of all finite linear combinations of bilinear forms above. On $\mathcal{C}$, we can define the following inner product:

$$(u \otimes v,\ w \otimes z)_{\mathcal{C}} := (u, w)_{\mathcal{H}_1}(v, z)_{\mathcal{H}_2} \qquad \forall\, u, w \in \mathcal{H}_1,\ \forall\, v, z \in \mathcal{H}_2. \tag{2.5}$$

Inner product (2.5) is well-defined and positive definite (see [111, Proposition 1, Section II.4]). Then, the *tensor product* $\mathcal{H}_1 \otimes \mathcal{H}_2$ of Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ is defined as the completion of the space $\mathcal{C}$ under inner product (2.5). In particular, if $\{u_n\}_{n\in\mathbb{N}}$ and $\{v_m\}_{m\in\mathbb{N}}$ are the orthonormal basis of $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, then $\{u_n \otimes v_m\}_{n,m\in\mathbb{N}}$ is an orthonormal basis of $\mathcal{H}_1 \otimes \mathcal{H}_2$ (see [111, Proposition 2, Section II.4]).

Now, let (T, Σ, μ) be a measure space and $\{\psi_m\}_{m\in\mathbb{N}}$ be the orthonormal basis of a separable Hilbert space $\mathcal{H}$ with inner product $(\cdot,\cdot)_{\mathcal{H}}$. We recall the following important result which will be used when describing the discretisation of PDEs with parametric inputs (see Section 4.2). The Lebesgue-Bochner space $L^2_\mu(T;\mathcal{H})$ (see (2.4)) can be uniquely identified with the tensor product space $L^2_\mu(T) \otimes \mathcal{H}$. In fact, for all functions $v \in L^2_\mu(T;\mathcal{H})$, we have in $\mathcal{H}$,

$$v(t) = \lim_{M\to\infty} \sum_{m=1}^{M} f_m(t)\psi_m \qquad \text{with} \qquad f_m(t) := (\psi_m, v(t))_{\mathcal{H}} \in L^2_\mu(T).$$

Then, there exists a unique isomorphism $U : L^2_\mu(T) \otimes \mathcal{H} \to L^2_\mu(T;\mathcal{H})$ such that $f(t) \otimes \psi \mapsto f(t)\psi$ for all $f \in L^2_\mu(T)$ and $\psi \in \mathcal{H}$, i.e. (see [111, Theorem II.10] or [118, Theorem B.17 and Remark C.24]),

$$L^2_\mu(T) \otimes \mathcal{H} \overset{U}{\cong} L^2_\mu(T;\mathcal{H}). \tag{2.6}$$

2.2 Discrete function spaces

The discretisation of domains as well as the (mesh-)refinement strategy are two fundamental ingredients of the design and the implementation of adaptive algorithms for PDEs in the context of the finite element method (FEM). In this section, we recall the notion of triangulation of bounded domains and define the corresponding discrete function spaces on such triangulations.

2.2.1 Triangulations

Triangulations of bounded domains rely on the definition of d-simplices. These are d-dimensional objects defined as the convex hull of $(d+1)$ points. Specifically, let $d \in \mathbb{N}$ and S be a set of $(d+1)$ affinely independent points $\mathbf{x}^{(1)},\dots,\mathbf{x}^{(d+1)}$ in $\mathbb{R}^d$. We say that $T \subset \mathbb{R}^d$ is the *d-simplex* generated by the points of S if

$$T = \mathrm{conv}(S) := \left\{ \mathbf{x} = \sum_{i=1}^{d+1} \lambda_i \mathbf{x}^{(i)} \ : \ \mathbf{x}^{(i)} \in S,\ 0 \le \lambda_i \le 1 \ \ \forall\, i = 1,\dots,d+1, \ \text{and} \ \sum_{i=1}^{d+1} \lambda_i = 1 \right\}.$$

The points $\mathbf{x}^{(1)},\dots,\mathbf{x}^{(d+1)}$ are called the *vertices* (or *nodes*) of T. For instance, a 1-simplex is a segment, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron. For any d-simplex T and a non-negative integer $m \le d-1$, we can further define the *m-dimensional face* of T as the m-simplex generated by the $m+1$ points of T. In particular, 0-dimensional faces are simple points, 1-dimensional faces are called the *edges* and 2-dimensional faces are called the *facets* of T.

Let us now introduce the notion of triangulation of a domain; see, e.g., [42, 35].

Definition 2.3 (Conforming triangulation)**.** *Let $D \subset \mathbb{R}^d$, $d = 2,3$, be a polygonal or polyhedral domain. A finite set $\mathcal{T}$ is a conforming*[1] *triangulation (or mesh) of D if it fulfils:*

- *each element $T \in \mathcal{T}$ is a d-simplex generated by $(d+1)$ vertices $x_T^{(1)},\dots,x_T^{(d+1)} \in \overline{D}$;*
- *the union of all elements of $\mathcal{T}$ covers the closure of the domain, i.e., $\overline{D} = \bigcup_{T\in\mathcal{T}} T$;*
- *the intersection of two elements $T, T' \in \mathcal{T}$ can be either empty, a vertex, or an edge (resp. facet) if $d = 2$ (resp. $d = 3$).*

The third condition of Definition 2.3 guarantees that a conforming triangulation $\mathcal{T}$ does not contain any *hanging node*, i.e., a vertex of an element $T \in \mathcal{T}$ which is contained in the interior of an edge (or facet) of some element $T' \in \mathcal{T}$.

Given $\mathcal{T}$, we denote by $\mathcal{N}(\mathcal{T}) := \bigcup_{T\in\mathcal{T}} \mathcal{N}(T)$ the set of vertices of $\mathcal{T}$, where $\mathcal{N}(T)$ denotes the set of vertices of $T \in \mathcal{T}$. Analogously, the set of edges (resp. facets) is given by $\mathcal{E}(\mathcal{T}) := \bigcup_{T\in\mathcal{T}} \mathcal{E}(T)$, with $\mathcal{E}(T)$ denoting the set of edges (resp. facets) of $T \in \mathcal{T}$. Furthermore, we denote by $\mathcal{N}^\circ(\mathcal{T}) \subset \mathcal{N}(\mathcal{T})$ the set of *interior* vertices, i.e., $\mathbf{x} \in \mathcal{N}^\circ(\mathcal{T})$ if and only if $\mathbf{x} \notin \partial D$, and we denote by $\mathcal{E}^\circ(\mathcal{T})$ the set of interior edges (or facets), i.e., $E \in \mathcal{E}^\circ(\mathcal{T})$ if and only if $E = T \cap T'$ for two elements $T, T' \in \mathcal{T}$.

[1] Conforming triangulations are also referred to as *admissible* triangulations; see, e.g., [34].

We introduce some further notation. The following sets

$$\omega(T) := \bigcup \Big\{ T' \in \mathcal{T} \ : \ \mathcal{E}(T) \cap \mathcal{E}(T') \neq \emptyset \Big\} \quad \text{and} \quad \omega(\mathsf{x}) := \bigcup \Big\{ T \in \mathcal{T} \ : \ \mathsf{x} \in \mathcal{N}(T) \Big\}, \tag{2.7}$$

are the *element patch* and the *vertex patch* of an element $T \in \mathcal{T}$ and a vertex $\mathsf{x} \in \mathcal{N}(\mathcal{T})$, respectively. Furthermore, define

$$h_T := \sup_{\mathsf{x},\mathsf{x}' \in T} |\mathsf{x} - \mathsf{x}'| \qquad \text{and} \qquad \rho_T := 2 \sup \Big\{ r > 0 \ : \ B(\mathsf{x}, r) \subset T, \, \mathsf{x} \in T \Big\} \quad \forall \ T \in \mathcal{T}, \tag{2.8}$$

where $B(\mathsf{x}, r)$ denotes the d-dimensional ball of radius r and centre x, and let $h := \max_{T \in \mathcal{T}} h_T$ be the *mesh-size* of $\mathcal{T}$. We say that a triangulation $\mathcal{T}$ is *shape-regular* if there exists a positive constant C_1 such that

$$\sigma(\mathcal{T}) := \max_{T \in \mathcal{T}} \sigma(T) \ \leq \ C_1 \qquad \text{with} \qquad \sigma(T) := \frac{h_T}{\rho_T} \qquad \forall \ T \in \mathcal{T}, \tag{2.9}$$

where $\sigma(T)$ is the *shape-regularity constant* which controls the degeneracy of an element $T \in \mathcal{T}$. In addition, a triangulation $\mathcal{T}$ is said to be *quasi-uniform* if all elements are of comparable size, i.e., there exists a positive constant C_2 such that

$$\frac{\max_{T \in \mathcal{T}} |T|}{\min_{T \in \mathcal{T}} |T|} \leq C_2. \tag{2.10}$$

Notice that if a triangulation is quasi-uniform, then it is also shape-regular, but not conversely. Furthermore, we say that a given sequence of triangulations $(\mathcal{T}_\ell)_{\ell \in \mathbb{N}}$ is shape-regular (resp. quasi-uniform) if (2.9) (resp. (2.10)) holds for all triangulations $\mathcal{T}_\ell$ ($\ell \in \mathbb{N}$).

2.2.2 Piecewise polynomial spaces

We now introduce the discrete function spaces on triangulations of domains $D \subset \mathbb{R}^d$, $d = 2, 3$. To this end, for each $k \in \mathbb{N}_0$, let $\mathcal{P}_k$ denote the space of polynomials of total degree less than or equal to k in the d variables $x_1, \ldots, x_d$. It is well known that $\dim(\mathcal{P}_k) = (k+d)!/(k!d!)$ for general $d \in \mathbb{N}$ (see, e.g., [41, Theorem 2, p. 29]).

For a triangulation $\mathcal{T}$ of D, we define the *finite element space* on D as the space of globally continuous piecewise polynomials of $\mathcal{P}_k$ over each element $T \in \mathcal{T}$, i.e.,

$$\mathcal{S}^k(\mathcal{T}) := \Big\{ v \in C^0(\overline{D}) \ : \ v|_T \in \mathcal{P}_k(T), \ \forall \ T \in \mathcal{T} \Big\} \qquad \forall \ k \in \mathbb{N}, \tag{2.11}$$

with inclusion $\mathcal{S}^k(\mathcal{T}) \subset H^1(D)$ for all $k \in \mathbb{N}$ (see, e.g., [42, Theorem 2.1.1]). For functions vanishing

on the boundary of D we define

$$\mathcal{S}_0^k(\mathcal{T}) := \left\{ v \in \mathcal{S}^k(\mathcal{T}) \ : \ v|_{\partial D} = 0 \right\} \subset H_0^1(D). \tag{2.12}$$

A basis for the space (2.11) is easily constructed by considering the $\#\mathcal{N}(\mathcal{T})$ Lagrange basis functions $\varphi_j \in \mathcal{S}^k(\mathcal{T})$ such that $\varphi_j(\mathsf{x}_i) = \delta_{ij}$, with δ_{ij} denoting the Kronecker symbol and x_i being a vertex of the triangulation $\mathcal{T}$, $i = 1, \ldots, \#\mathcal{N}(\mathcal{T})$ (here, # denotes the cardinality); see, e.g., [42] and [35, Section 3.1]. Hence, $\dim(\mathcal{S}^k(\mathcal{T})) = \#\mathcal{N}(\mathcal{T})$. Analogously, a basis for (2.12) would only include those basis elements φ_j associated with interior vertices $\mathsf{x}_j \in \mathcal{N}^\circ(\mathcal{T})$. The functions φ_j are often referred to as *nodal* basis functions and for $k = 1$, the are typically called *hat* functions.

2.3 Adaptive mesh-refinement in the finite element setting

One of the main topics of this work is the design of adaptive algorithms for (parametric) problems in the setting of the finite element method. For a given model problem, e.g., a PDE posed on a bounded domain where spatial discretisations are made via finite element spaces on conforming triangulations (see (2.11) and (2.12)), adaptive FEM algorithms typically follow the standard loop consisting of the following four modules

$$\mathsf{SOLVE} \quad \Longrightarrow \quad \mathsf{ESTIMATE} \quad \Longrightarrow \quad \mathsf{MARK} \quad \Longrightarrow \quad \mathsf{REFINE}. \tag{2.13}$$

The module SOLVE computes a numerical solution of the problem under consideration whereas the module ESTIMATE usually gives information about the distribution of the estimated error among the elements of the underlying triangulation. Then, in order to construct enhanced approximations for the sake of reducing the error, the module MARK implements some marking strategy to select the elements associated with comparably large errors and the module REFINE comes with some refinement rule which determines the refinement of such elements. For an introduction to the finite element method we refer to, e.g., [42, 34, 35, 125]; comprehensive reviews of the aspects of adaptive FEMs may be found in, e.g., [104, 105, 121].

In what follows, we recall two popular marking strategies that are often employed in the MARK module of adaptive finite element algorithms and we also describe the mesh-refinement technique that will be used in the REFINE module of all adaptive algorithms presented in this thesis.

Maximum marking strategy

Input: *set* $\{\beta(S)\}_{S\in\mathcal{S}}$ *and a marking parameter* $\theta \in (0,1]$.
DO
    set $\beta_{\max} := \max\{\beta(S) \,:\, S \in \mathcal{S}\}$;
END
FOR $S \in \mathcal{S}$
    IF $\beta(S) \geq \theta\, \beta_{\max}$
        set $\mathcal{M} = \mathcal{M} \cup \{S\}$;
    END
END
Output: *subset of marked elements* $\mathcal{M} \subseteq \mathcal{S}$.

Strategy 2.1. The maximum marking strategy with threshold parameter θ for a given input set of numbers $\{\beta(S)\}$ associated with the elements S of a generic set $\mathcal{S}$.

2.3.1 Marking strategies

Suppose that a set of error estimates $\{\beta(T)\}_{T\in\mathcal{T}}$ associated with elements $T \in \mathcal{T}$ is available; in an adaptive FEM algorithm these are typically computed by some (*a posteriori*) error estimation technique implemented in the ESTIMATE module (see, e.g., [2, 131]). Two popular marking strategies to select a subset of elements which contribute with large errors are the *maximum* marking strategy and the *Dörfler* marking strategy[2].

Early uses of the maximum marking strategy date back to [10]. In this strategy, an element $T \in \mathcal{T}$ is marked if the associated error estimate $\beta(T)$ is larger than a fixed proportion of the maximum of all error estimates. That is, for a given marking (or *threshold*) parameter $\theta \in (0,1]$, the strategy returns a minimal set $\mathcal{M} \subseteq \mathcal{T}$ of marked elements such that

$$\beta(T) \;\geq\; \theta \max_{T\in\mathcal{T}} \beta(T) \qquad \forall\, T \in \mathcal{M}. \tag{2.14}$$

Notice that large values of θ lead to small subsets of marked elements and vice versa.

The Dörfler marking strategy, introduced in [52], builds a subset of marked elements $\mathcal{M} \subseteq \mathcal{T}$ with minimal cardinality satisfying

$$\sum_{T\in\mathcal{M}} \beta(T)^2 \;\geq\; \theta \sum_{T\in\mathcal{T}} \beta(T)^2, \tag{2.15}$$

where $\theta \in (0,1]$ is the associated marking parameter. Contrary to the maximum strategy, here,

[2] The Dörfler marking strategy may be also referred to as *equilibration* (see, e.g., [131]) or *bulk chasing* (see, e.g., [105]) strategy.

Dörfler marking strategy

Input: *set* $\{\beta(S)\}_{S\in\mathcal{S}}$ *and a marking parameter* $\theta \in (0,1]$.
DO
set $\beta(\mathcal{S})^2 := \sum_{S\in\mathcal{S}} \beta(S)^2$, $\mathcal{M} = \emptyset$, *and* $\beta(\mathcal{M}) := 0$;
END
WHILE $\beta(\mathcal{M})^2 < \theta\,\beta(\mathcal{S})^2$
set $\beta_{\max} := \max\{\beta(S)^2 \ : \ S \in \mathcal{S}\setminus\mathcal{M}\}$;
FOR $S \in \mathcal{S}\setminus\mathcal{M}$
IF $\beta(S)^2 = \beta_{\max}$
set $\beta(\mathcal{M})^2 = \beta(\mathcal{M})^2 + \beta(S)^2$ *and* $\mathcal{M} = \mathcal{M}\cup\{S\}$;
END
END
END
Output: *subset of marked elements* $\mathcal{M} \subseteq \mathcal{S}$.

Strategy 2.2. The Dörfler marking strategy with threshold parameter θ for a given input set of numbers $\{\beta(S)\}$ associated with the elements S of a generic set $\mathcal{S}$.

large values of θ lead to large subsets of marked elements and it is guaranteed that sufficiently many elements are selected so that their combined contributions to the total error estimate constitutes a fixed proportion thereof. Notice that in order to construct a minimal subset, the sum on the left hand-side of (2.15) considers the elements according to the descendent magnitudes of associated estimates. This way, the set $\{\beta(T)\}_{T\in\mathcal{M}}$ satisfying (2.15) consists of the $\#\mathcal{M}$ largest error estimates of the full set $\{\beta(T)\}_{T\in\mathcal{T}}$.

The maximum strategy is listed in Strategy 2.1 whereas the Dörfler strategy is listed in Strategy 2.2. In both cases, the strategies are listed for generic quantities $\{\beta(S)\}$ associated with the elements S of a set $\mathcal{S}$.

2.3.2 Newest vertex bisection

The *newest vertex bisection* (NVB) (see [120]) is widely used in the FEM context since it turned out to be a key ingredient for proving the convergence of adaptive algorithms for the numerical approximation of solutions to PDEs; see, e.g., [52, 97, 31, 93, 40].

For simplicity, let us consider $D \subset \mathbb{R}^2$. Let $\mathcal{T}_0$ be an initial triangulation of D. In the NVB, a *reference* edge is chosen for each element $T \in \mathcal{T}_0$. This can be, for example, the longest edge of each element (see [113]). Given a set of elements $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ to be refined (obtained by employing,

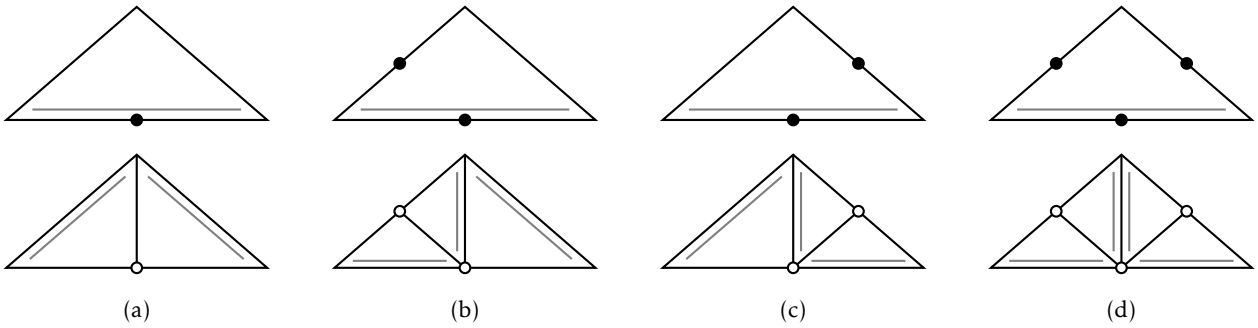


Figure 2.1. NVB bisections. (a) One, (b)-(c) two, and (d) three bisections of the edges of an element in $\mathcal{T}_\ell$. Double lines denote the reference edges, black dots denote the edges to be bisected, and white dots denote the newest vertices. Reference edges are always bisected first.

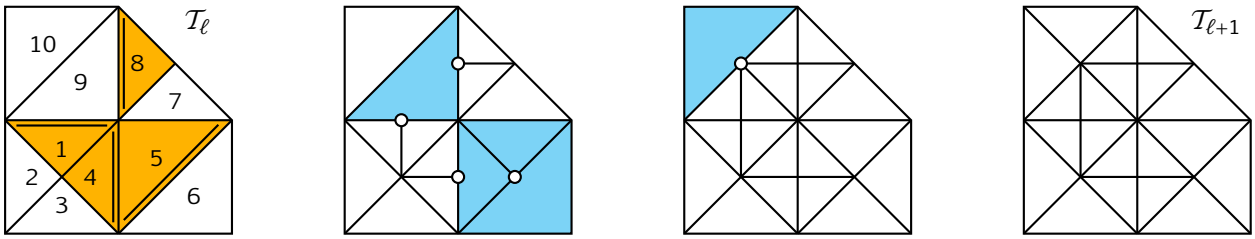
e.g., either the maximum or the Dörfler marking strategy), for $\ell \in \mathbb{N}_0$, successive iterations of NVB refinements read as follows:

1) for each elements $T \in \mathcal{M}_\ell$, the midpoint z_T of the reference edge is connected with the vertex of T in front of the reference edge. Then, z_T becomes a new vertex;

2) such bisection (for all $T \in \mathcal{M}_\ell$) produces two new elements (the *children* of T) whose reference edges are the edges in front of the newest vertex z_T; see Figure 2.1(a);

3) the triangulation obtained by the refinement of every marked element is not usually conforming due to the presence of hanging nodes. Hence, additional bisections are required to yield $\mathcal{T}_{\ell+1}$.

The third step of the NVB iteration described above is referred to as *completion*, or *mesh-closure*, step. Notice that hanging nodes appear in the interior of those edges $E = T_1 \cap T_2$, with $T_1, T_2 \in \mathcal{T}_\ell$, for which the refinement of only one between T_1 and T_2 involves the bisection of E. To get rid of hanging nodes, the algorithm performs iterated newest vertex bisections which can split an element $T \in \mathcal{T}_\ell$ into two, three, or four new elements; see Figure 2.1. Observe that according to the configuration of $\mathcal{T}_\ell$, these additional bisections are likely to involve elements which do not belong to the set of marked elements; see Figure 2.2. It is easy to see that the completion step consists of finitely many additional bisections since the NVB performs at most $3\#\mathcal{T}_\ell$ bisections in case all edges of $\mathcal{T}_\ell$ are bisected. Furthermore, we emphasise that since the refinement of an element involves the bisection of some of its edges, NVB iterations are equivalently defined for a given input set of marked edges $\mathcal{M}_\ell \subseteq \mathcal{E}(\mathcal{T}_\ell)$. For additional details, we refer to [17, 84, 124] for NVB refinements in two and three-dimensional cases and to [94] for an overview and comparison of the NVB with other different mesh-refinement techniques.



(a) (b) (c) (d)

Figure 2.2. NVB refinement. (a) Triangulation $\mathcal{T}_\ell$ with marked elements $\mathcal{M}_\ell = \{T_1, T_4, T_5, T_8\}$ in orange. Double lines denote the reference edges of marked elements; (b) Bisections of reference edges of marked elements $\mathcal{M}_\ell$. Four newest vertices (white dots), which are hanging nodes, are introduced; (c)-(d) Completion steps. Blue elements represent the elements of $\mathcal{T}_\ell$ that require further bisections to yield $\mathcal{T}_{\ell+1}$. The set of overall elements that are refined to obtain $\mathcal{T}_{\ell+1}$ is $\mathcal{R}_\ell := \mathcal{M}_\ell \cup \{T_6, T_9, T_{10}\}$. In particular, there has been one bisection in elements T_1, T_4, T_6, T_8, and T_{10}, two bisections in element T_5, and three bisections in element T_9.

Hereafter, for any given conforming triangulation $\mathcal{T}$ of D, we will let $\mathsf{REFINE}(\cdot)$ be any subroutine implementing NVB refinements such that $\widetilde{\mathcal{T}} = \mathsf{REFINE}(\mathcal{T}, \mathcal{M})$ returns the coarsest conforming triangulation $\widetilde{\mathcal{T}}$ so that all elements (resp. edges) in $\mathcal{M}$ have been refined (resp. bisected). In particular, $\mathcal{T} = \mathsf{REFINE}(\mathcal{T}, \emptyset)$. For each $n \in \mathbb{N}$, a triangulation $\mathcal{T}_n$ is a refinement of $\mathcal{T}_0$ if $\mathcal{T}_n$ can be obtained by a finite sequence of refinements $\mathcal{T}_{\ell+1} = \mathsf{REFINE}(\mathcal{T}_\ell, \mathcal{M}_\ell)$ for all $\ell = 0, \dots, n-1$. Moreover, in this work, we say that $\widehat{\mathcal{T}}$ is a *uniform* refinement of $\mathcal{T}$ if $\widehat{\mathcal{T}}$ is obtained by three newest vertex bisections per each element $T \in \mathcal{T}$ (see Figure 2.1(d)).

Remark 2.3.1. *An important feature of NVB refinements is that they automatically lead to nested finite element spaces. That is, for some $\ell \in \mathbb{N}_0$, if $\mathcal{S}^k(\mathcal{T}_\ell)$ denotes the finite element space* (2.11) *associated with triangulation $\mathcal{T}_\ell$, after one NVB refinement, the larger finite element space $\mathcal{S}^k(\mathcal{T}_{\ell+1})$ constructed on $\mathcal{T}_{\ell+1}$ is such that $\mathcal{S}^k(\mathcal{T}_\ell) \subseteq \mathcal{S}^k(\mathcal{T}_{\ell+1})$; see, e.g., [105]. This is not guaranteed by other mesh-refinement techniques, such as red-green refinements (see [14]) and red-green-blue refinements (see [130]).*

Chapter 3

Random fields

In many practical applications, the main challenge is to cope with the uncertainty in some physical quantity of the model which may vary either in space or in time. A good characterisation of such uncertainty is therefore essential and represents the reason for developing efficient numerical methods for reliable uncertainty quantification. At the basis of effective mathematical modelling of uncertain inputs in the model's data there are probability spaces and the notion of random variable. In this work, we are only concerned with model problems whose responses have uncertain spatial behaviour. In this case, spatially varying random fields represent the main tool to model and handle the uncertainties numerically.

In this chapter, we first recall the concepts of probability space and random variable staying within the classical formalism of probability theory (see, e.g., [86, 19]). Then, we address the problem of the representation of random fields by describing the Karhunen-Loève (KL) expansions and, among spectral methods, we briefly review the idea of (generalised) Polynomial Chaos (PC) expansions of random fields.

3.1 Random variables and probability spaces

A *probability space* is a measure space $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$ where Ω is a non-empty set called the *sample space*, $\mathcal{F}(\Omega)$ is a σ-algebra on Ω, and $\mathbb{P}$ is a *probability measure*, i.e., a measure $\mathbb{P} : \Omega \to [0,1]$ such that $\mathbb{P}(\Omega) = 1$. A $(\Psi, \mathcal{F}(\Psi))$-*random variable* Y on $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$ is a measurable function from $(\Omega, \mathcal{F}(\Omega))$ to $(\Psi, \mathcal{F}(\Psi))$. The observed value $y := Y(\omega) \in \Psi$, for some $\omega \in \Omega$, is called *realisation* of Y. We say that Y is *real valued* when it takes values in $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, with $\mathcal{B}(\mathbb{R})$ being the Borel σ-algebra on $\mathbb{R}$.

For any real-valued random variable Y on $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$ there is a *probability distribution* $\mathbb{P}_Y$

defined as the probability measure on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ such that

$$\mathbb{P}_Y(B) := \mathbb{P}(Y^{-1}(B)) = \mathbb{P}\Big(\{\omega \in \Omega \,:\, Y(\omega) \in B\}\Big) \qquad \forall\, B \in \mathcal{B}(\mathbb{R}). \tag{3.1}$$

If $\mathbb{P}_Y$ is absolutely continuous with respect to the Lebesgue measure $d\mathsf{x}$, then there exists a *probability density* (function) $\rho_Y : \mathbb{R} \to [0, +\infty)$ such that

$$\mathbb{P}_Y(B) = \int_B \rho_Y(y)\, dy \qquad \forall\, B \in \mathcal{B}(\mathbb{R}). \tag{3.2}$$

Notice that $\int_{\mathbb{R}} \rho_Y(y)\, dy = 1$, since $\mathbb{P}_Y(\mathbb{R}) = \mathbb{P}(Y^{-1}(\mathbb{R})) = \mathbb{P}(\Omega) = 1$ as $\mathbb{P}$ is a probability measure. The *mean* and the *variance* of Y are defined by

$$\mathbb{E}[Y] := \int_\Omega Y(\omega)\, d\mathbb{P}(\omega) = \int_{\mathbb{R}} y\, d\mathbb{P}_Y(y) = \int_{\mathbb{R}} y\, \rho_Y(y)\, dy, \tag{3.3}$$

$$\mathsf{Var}(Y) := \mathbb{E}\Big[\,(Y - \mathbb{E}[Y])^2\,\Big] = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2, \tag{3.4}$$

respectively, where equalities in (3.3) hold due to (3.1) and (3.2). The *standard deviation* of Y is the quantity defined by $\sigma_Y := \sqrt{\mathsf{Var}(Y)}$.

Now, let $\Gamma_m \subset \mathbb{R}$ for all $m \in \mathbb{N}$, and consider a sequence $(Y_m)_{m\in\mathbb{N}}$ of $(\Gamma_m, \mathcal{B}(\Gamma_m))_{m\in\mathbb{N}}$-random variables on $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$. Let $(\mathbb{P}_{Y_m})_{m\in\mathbb{N}}$ be the associated probability distributions. We denote by $\Gamma := \bigotimes_{m\in\mathbb{N}} \Gamma_m$ the *product space* of $(\Gamma_m)_{m\in\mathbb{N}}$ and by $\mathcal{B}(\Gamma) := \bigotimes_{m\in\mathbb{N}} \mathcal{B}(\Gamma_m)$ the *product Borel σ-algebra* defined as the smallest Borel σ-algebra containing all sets $\bigotimes_{m\in\mathbb{N}} B_m$, with $B_m \in \mathcal{B}(\Gamma_m)$ for all $m \in \mathbb{N}$. The *product probability measure* $\otimes\mathbb{P}$ on $(\Gamma, \mathcal{B}(\Gamma))$ is defined as the unique (see, e.g., [19, Theorem 9.2]) probability measure such that

$$\otimes\mathbb{P}\left(\bigotimes_{m\in\mathbb{N}} B_m\right) = \prod_{m\in\mathbb{N}} \mathbb{P}_{Y_m}(B_m) \qquad \forall\, B_m \in \mathcal{B}(\Gamma_m). \tag{3.5}$$

Let $\boldsymbol{Y}(\omega) = (Y_m(\omega))_{m\in\mathbb{N}}$, $\omega \in \Omega$, be a multivariate random variable with joint probability distribution $\mathbb{P}_{\boldsymbol{Y}}(B) := \mathbb{P}(\{\omega \in \Omega \,:\, \boldsymbol{Y}(\omega) \in B\})$ for all $B \in \mathcal{B}(\Gamma)$ (cf. (3.1)). One can show that this is a well-defined measure on $(\Gamma, \mathcal{B}(\Gamma))$. Also, $\rho_{\boldsymbol{Y}}$ represents the associated *joint probability density*. We say that random variables $(Y_m)_{m\in\mathbb{N}}$ are *independent* if the joint distribution $\mathbb{P}_{\boldsymbol{Y}}$ is equal to the product measure $\otimes\mathbb{P}$ defined in (3.5), or, equivalently, if $\rho_{\boldsymbol{Y}} = \prod_{m\in\mathbb{N}} \rho_{Y_m}$, where ρ_{Y_m} are the probability densities of Y_m for all $m \in \mathbb{N}$ (see (3.2)). In particular, this implies that

$$\mathbb{E}[\boldsymbol{Y}] = \int_\Gamma \mathbf{y}\, \rho_{\boldsymbol{Y}}(\mathbf{y})\, d\mathbf{y} = \prod_{m\in\mathbb{N}} \int_{\Gamma_m} y_m \rho_{Y_m}(y_m) dy_m = \prod_{m\in\mathbb{N}} \mathbb{E}[Y_m] \qquad \text{where} \qquad \mathbf{y} := \boldsymbol{Y}(\omega).$$

In addition, if the random variables are independent and, for all $m \neq n$, there holds $\rho_{Y_m} = \rho_{Y_n}$, we say that they are *independent and identically distributed* (i.i.d.), whereas we say that Y_m and Y_n are

uncorrelated if $\mathbb{E}[Y_m Y_n] = 0$ for all $m \neq n$.

3.2 Definition of random field

Random fields are very important in many engineering applications as they are typically used to define or obtain other quantities of the model by means of, for example, a PDE. These quantities can be velocity fields, temperatures, pressures, etc., according to the model under consideration. However, in all those problems whose inputs are uncertain or depend on some random parameters, one does not have in general the exact knowledge or representation of the random field itself. We can only rather obtain a priori information from some physical property or quantity which is relevant to the application. It is therefore essential to be able to construct or approximate random fields appropriately.

A formal definition of a random field can be given as follows. Let $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$ be a probability space and $D \subset \mathbb{R}^d$, $d \in \mathbb{N}$, be a bounded spatial domain. A *random field* is a jointly measurable function $a : D \times \Omega \to \mathbb{R}$ with respect to $\mathcal{F}(\Omega)$ on the sample space Ω and the Borel σ-algebras on D and $\mathbb{R}$. In particular, a random field a can be seen as the family $\{a(\mathbf{x}, \omega)\}_{\mathbf{x}\in D,\, \omega\in\Omega}$ such that $a(\mathbf{x}, \cdot)$ is a random variable on $(\Omega, \mathcal{F}(\Omega))$ for any fixed point $\mathbf{x} \in D$ and $a(\cdot, \omega)$ is a realisation in D for any fixed $\omega \in \Omega$; see, e.g., [88].

Analogously to random variables, for all $\omega \in \Omega$, we can define the following quantities

$$\mathbb{E}[a](\mathbf{x}) := \int_\Omega a(\mathbf{x}, \omega)\, d\mathbb{P}(\omega) \qquad \forall\, \mathbf{x} \in D, \tag{3.6}$$

$$\mathrm{Cov}[a](\mathbf{x}, \mathbf{x}') := \mathbb{E}\Big[\big(a(\mathbf{x}, \omega) - \mathbb{E}[a](\mathbf{x})\big)\big(a(\mathbf{x}', \omega) - \mathbb{E}[a](\mathbf{x}')\big)\Big] \qquad \forall\, \mathbf{x}, \mathbf{x}' \in D, \tag{3.7}$$

which are the mean and the *covariance* of the random field a, respectively. The *variance* of a is defined as $\mathrm{Var}(a)(\mathbf{x}) := \mathrm{Cov}[a](\mathbf{x}, \mathbf{x})$ for all $\mathbf{x} \in D$. Note that for these quantities to be well-defined we should assume that the random field is *second-order*, i.e., a belongs to the Lebesgue-Bochner space $L^2_{\mathbb{P}}(\Omega; L^2(D))$. Furthermore, we say that $\mathrm{Cov}[a]$ is *positive definite* if

$$\sum_{m\in\mathbb{N}} \sum_{n\in\mathbb{N}} c_m \mathrm{Cov}[a](\mathbf{x}_m, \mathbf{x}_n)\overline{c}_n \geq 0 \qquad \forall\, \mathbf{x}_m, \mathbf{x}_n \in D,\ c_m, c_n \in \mathbb{C}. \tag{3.8}$$

Before addressing the problem of the representation of random fields, below we give two examples of important classes of random fields which we will use in some numerical experiments in this thesis.

Example 3.2.1 (Stationary random fields)**.** *We say that a second-order random field is stationary, if its mean* $\mathbb{E}[a](\mathbf{x})$ *is constant (i.e., it is independent of* $\mathbf{x} \in D$*) and the covariance can be written as* $\mathrm{Cov}[a](\mathbf{x}, \mathbf{x}') = c(\mathbf{x} - \mathbf{x}')$ *for some function* $c : D \to \mathbb{R}$ *called stationary covariance function. That is, stationary random fields are invariant to translations. On simple two-dimensional rectangular domains* $D = [-b_1, b_1] \times [-b_2, b_2]$ *with* $[-b_k, b_k] \subset \mathbb{R}$*,* $k = 1, 2$*, a typical example of a stationary covariance function is the following,*

$$\mathrm{Cov}[a](\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{|x_1 - x_1'|}{\ell_1} - \frac{|x_2 - x_2'|}{\ell_2}\right), \tag{3.9}$$

where $\mathbf{x} = (x_1, x_2), \mathbf{x}' = (x_1', x_2') \in D$*, and* $\ell_1, \ell_2 > 0$ *are the correlation lengths; see [88, Example 7.56].*

Example 3.2.2 (Isotropic random fields)**.** *We say that a second-order random field is isotropic if it is stationary and its stationary covariance function* $c : D \to \mathbb{R}$ *is given by* $c(\mathbf{x}) = c_0(r)$*, where* $r := \sqrt{x_1^2 + x_2^2}$ *and* c_0 *is called the isotropic covariance function. That is, isotropic random fields are stationary random fields invariant to rotations. A typical example of an isotropic covariance function on two-dimensional domains is given by*

$$c_0(\mathbf{x}) = \frac{1}{4\ell^2} \exp\left(-\frac{\pi r^2}{4\ell^2}\right), \qquad \mathbf{x} \in D, \tag{3.10}$$

for a correlation length $\ell > 0$*; see [88, Example 6.10].*

3.3 Representation of random fields

In order to take the uncertainty into account in a given model we need to represent random fields in an appropriate way. Typically, a random field can be written in Fourier-type series in which the spatial part, consisting of a family of real-valued functions, is separated from the stochastic part, which consists of a sequence of random variables. In this section, we briefly recall two well-known examples of such representations.

3.3.1 Karhunen-Loève expansions

The *Karhunen-Loève* (KL) expansion of a random field a is the preferred candidate in many applications since it represents an optimal approximation of a in the mean square sense when the corresponding infinite expansion is truncated after the first, say, $M \in \mathbb{N}$ terms.

Let $\mathcal{C}_a : L^2(D) \to L^2(D)$ be the *covariance operator* of a second-order random field a defined as

$$\mathcal{C}_a(f)(\mathbf{x}) := \int_D \mathrm{Cov}[a](\mathbf{x}, \mathbf{x}') f(\mathbf{x}')\, d\mathbf{x}' \qquad \forall\, f \in L^2(D). \tag{3.11}$$

If the symmetric covariance $\mathrm{Cov}[a]$ is positive definite (see (3.8)), then $\mathcal{C}_a$ is a symmetric, non-negative, and compact operator. In particular, there exists a family $(\lambda_m, f_m)_{m\in\mathbb{N}}$ of eigenpairs of $\mathcal{C}_a$, i.e., satisfying

$$\mathcal{C}_a(f_m) = \lambda_m f_m \qquad \forall\, m \in \mathbb{N}, \tag{3.12}$$

where the sequence of non-negative eigenvalues $(\lambda_m)_{m\in\mathbb{N}}$, which is enumerated according to decreasing magnitudes (i.e., $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$), tends to zero as $m \to \infty$, and eigenfunctions $(f_m)_{m\in\mathbb{N}}$ are orthonormal with respect to the inner product $(\cdot,\cdot)_{L^2(D)}$.

Definition 3.1 (KL-expansions)**.** *The Karhunen-Loève expansion of a second-order random field a is defined as*

$$a(\mathbf{x},\omega) = \mathbb{E}[a](\mathbf{x}) + \sum_{m=1}^{\infty} \sqrt{\lambda_m} f_m(\mathbf{x}) Y_m(\omega), \qquad \mathbf{x} \in D,\ \omega \in \Omega, \tag{3.13}$$

where $Y_m(\omega)$ are random variables uniquely defined by

$$Y_m(\omega) := \frac{1}{\sqrt{\lambda_m}} \left(a(\mathbf{x},\omega) - \mathbb{E}[a](\mathbf{x}),\, f_m(\mathbf{x}) \right)_{L^2(D)} \qquad \forall\, m \in \mathbb{N},\ \omega \in \Omega,$$

and $(\lambda_m, f_m)_{m\in\mathbb{N}}$ are the eigenpairs of the covariance operator $\mathcal{C}_a$ defined in (3.11).

It is straightforward to verify that Y_m are mean zero uncorrelated random variables with unit variance. Furthermore, expansion (3.13) is well-defined as it converges in $L^2_{\mathbb{P}}(\Omega; L^2(D))$ due to Mercer's theorem (see, e.g., [112, p. 245]). In fact, if we denote by

$$a_M(\mathbf{x},\omega) := \mathbb{E}[a](\mathbf{x}) + \sum_{m=1}^{M} \sqrt{\lambda_m} f_m(\mathbf{x}) Y_m(\omega), \qquad \mathbf{x} \in D,\ \omega \in \Omega, \tag{3.14}$$

the truncated KL-expansion of a after $M \in \mathbb{N}$ terms, then there holds

$$\sup_{\mathbf{x},\mathbf{x}'\in D} \left| \mathrm{Cov}[a](\mathbf{x},\mathbf{x}') - \mathrm{Cov}[a_M](\mathbf{x},\mathbf{x}') \right| = \sup_{\mathbf{x}\in D} \sum_{m=M+1}^{\infty} \lambda_m \varphi_m(\mathbf{x})^2 \to 0 \quad \text{as} \quad M \to \infty,$$

and then $\sup_{\mathbf{x}\in D} \mathbb{E}[(a - a_M)(\mathbf{x},\cdot)] = \sup_{\mathbf{x}\in D} \sum_{m=M+1}^{\infty} \lambda_m \varphi_m(\mathbf{x})^2 \to 0$ as $M \to \infty$ (see also [87, p. 144] and [88, Theorem 7.53]). In particular, the $L^2_{\mathbb{P}}(\Omega; L^2(D))$-error due to the truncation after M terms is then given by

$$\| a - a_M \|_{L^2_{\mathbb{P}}(\Omega;L^2(D))} = \sum_{m=M+1}^{\infty} \lambda_m. \tag{3.15}$$

This error is optimal in the sense that for any other truncated series $\widetilde{a}_M$ of $M \in \mathbb{N}$ random variables and spatial functions, error (3.15) due to KL-expansion is smaller than the corresponding error $\| a - \widetilde{a}_M \|_{L^2_{\mathbb{P}}(\Omega;L^2(D))}$ (see, e.g., [72]). It is clear that information about the decay of the eigenvalues $(\lambda_m)_{m\in\mathbb{N}}$ of $\mathcal{C}_a$ plays a crucial role in order to obtain good bounds which control the error of

truncated KL expansions; see, e.g., [68, 119].

From the description above, we see that KL-expansions typically require the (a priori) knowledge of prescribed means $\mathbb{E}[a]$ and covariances $\mathrm{Cov}[a]$. In addition, if the eigenpairs of the associated covariance operator $\mathcal{C}_a$ are not given/known, they should be computed by solving (3.12) which is in general a non-trivial task. For example, discretisation of (3.12) obtained from numerical methods such as Galerkin projections and approximations via collocation points, give rise to matrix equations whose eigenpairs approximate the eigenpairs of the continuous problem (see, e.g., [88, Section 7.4]). However, such resulting matrices are typically dense and large, thus efficient and accurate numerical approximations may result in expensive computations. In particular cases, such as exponential (see (3.9)) or *triangular* covariances on certain domains, there exist the analytical expressions of the corresponding eigenpairs (see, e.g., [72, Chapter 2] and Example 3.3.1 below). For other general covariances or complicated domains, one may use efficient eigensolvers, e.g., so called fast multipole methods (see [119]).

Example 3.3.1 (Stationary covariances)**.** *Consider the KL-expansion of a random field with the stationary exponential covariance* (3.9) *from Example 3.2.1. In this case, the eigenpairs* $(\lambda_m, f_m)_{m\in\mathbb{N}}$ *of the covariance operator* $\mathcal{C}_a$ *of the associated random field are given by* $f_m(\mathbf{x}) := f_i^{(1)}(x_1) f_j^{(2)}(x_2)$ *and* $\lambda_m := \lambda_i^{(1)}\lambda_j^{(2)}$*, where* $(\lambda_i^{(1)}, f_i^{(1)})_{i\in\mathbb{N}}$ *and* $(\lambda_j^{(2)}, f_j^{(2)})_{j\in\mathbb{N}}$ *are the eigenpairs of the one-dimensional eigenvalue problem*

$$\int_{-b_k}^{b_k} \exp\left(-\frac{|x-z|}{\ell_k}\right) f^{(k)}(z)dz = \lambda^{(k)} f(z), \qquad k = 1, 2. \tag{3.16}$$

See, e.g., [88, Example 7.55], for the analytical expression of such eigenpairs for problem (3.16)*. It can be shown that the values of correlation lengths does not affect the asymptotic decay (i.e., for* $m \to \infty$*) of eigenvalues* λ_m *which is of order* $\mathcal{O}(m^{-2})$ *(see [88, Example 7.58]). A truncated KL-expansion with covariance* (3.9) *will be considered in the numerical experiment in Section 7.3.2.*

In some cases, one may consider KL-expansions with *explicit* eigenpairs that yield random fields with covariance functions that are close/similar to goal covariances functions whose eigenpairs are either not known analytically or are too difficult to approximate.

Example 3.3.2 (Isotropic covariances)**.** *Let* $D = (0,1)^2$ *and consider the following KL-expansion of a random field* $a(\mathbf{x}, \omega)$*,*

$$a(\mathbf{x}, \omega) = \mathbb{E}[a](\mathbf{x}) + \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \sqrt{\nu_{ij}}\phi_{ij}(\mathbf{x})Y_{ij}(\omega), \quad \mathbf{x} \in D,\ \omega \in \Omega, \tag{3.17}$$

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

$$\phi_{ij}(\mathbf{x}) := 2\cos(i\pi x_1)\cos(j\pi x_2) \quad \textit{and} \quad \nu_{ij} := \frac{1}{4}\exp\left(-\pi(i^2+j^2)\ell^2\right) \quad \forall\, i,j \in \mathbb{N}, \tag{3.18}$$

with $\ell > 0$ *and random variables* Y_{ij} *being independent uniformly distributed in* $[-\sqrt{3},\sqrt{3}]$; *note that they are mean zero and have unit variance for all* $i,j \in \mathbb{N}_0$. *It can be shown that random field* $a(\mathbf{x},\omega)$ *in* (3.17) *has a covariance function close to the isotropic function* (3.10) *provided that* ℓ *is small enough; see [88, Example 9.37]. KL-expansion* (3.17) *will be considered in the numerical experiment in Section 5.4.4.*

Example 3.3.3 (Exponential random fields)**.** *The well-posedness of some model problem may require the input random field* a *to be positive, e.g.,* $a > a_{\min}$ *a.e. in* D *for some positive constant* $a_{\min}$. *To enforce such positiveness, we can represent* a *as a nonlinear function of the random variables. This is typically the case of exponential random fields of the form*

$$a(\mathbf{x},\omega) = a_{\min} + \exp\left(\sum_{m=1}^{\infty} a_m(\mathbf{x})Y_m(\omega)\right), \qquad \mathbf{x} \in D,\ \omega \in \Omega, \tag{3.19}$$

or, equivalently defined, e.g., as the KL-expansion of $\log(a - a_{\min})$ *assuming that* $a - a_{\min} > 0$ *almost surely. In particular, when* $(Y_m)_{m\in\mathbb{N}}$ *are normal random variables, random field* (3.19) *is called log-normal random field; see, e.g., [5, 103, 128, 101]. In this thesis we are not going to consider the case of exponential random fields, but we rather require the positiveness of the considered random fields by direct assumption.*

3.3.2 Polynomial Chaos expansions

The main drawback of KL-expansions is that they require the random field to have a known prescribed mean and covariance function. Furthermore, assuming such statistics for the input random field of the model under consideration does not guarantee at all the response to have the same properties, i.e., we cannot expect to represent the solution of the model via KL-expansions.

A more general approach to representing random fields dates back to Wiener [133] and was introduced for the representation of Gaussian random processes. This is known as *polynomial chaos* (PC) expansion. Here, random fields are represented by an infinite series in which the stochastic part consists of Hermite polynomials (see Example 4.2.2) in a sequence of independent Gaussian random variables. That is, the idea is to write a representation in which a polynomial combination of known random variables is used as a basis for the expansion. When random

variables are assumed to be not Gaussian, in order to work with orthogonal polynomials, Hermite polynomials are substituted by families of orthogonal polynomials with respect to the probability distribution of such random variables. In this case, we typically talk about *generalised* PC (gPC) expansions; see, e.g., [134, 135].

In general, for a sequence of independent real-valued $(\Gamma, \mathcal{B}(\Gamma))$-random variables $(Y_i(\omega))_{i\in\mathbb{N}}$ on the probability space $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$, with $\boldsymbol{Y}(\omega) = (Y_i(\omega))_{i\in\mathbb{N}}$, and a set $\{P_m\}_{m\in\mathbb{N}}$ of orthogonal polynomials in $\boldsymbol{Y}$ forming a basis of $L^2_{\mathbb{P}_{\boldsymbol{Y}}}(\Gamma)$, a second-order random field can then be expanded (i.e., written) as

$$a(\mathbf{x}, \omega) = \sum_{m=1}^{\infty} f_m(\mathbf{x}) P_m(\boldsymbol{Y}(\omega)), \qquad \mathbf{x} \in D,\ \omega \in \Omega, \tag{3.20}$$

where coefficients $f_m : D \to \mathbb{R}$ are spatial functions that have to be chosen appropriately; for example, they can be finite element functions; see, e.g., [72, 135]. There are many well-known families of gPC polynomial basis associated with the corresponding distribution of random variables $(Y_i)_{i\in\mathbb{N}}$. For example, Gaussian random variables lead to Hermite polynomials while uniformly distributed random variables lead to Legendre polynomials (see Example 4.2.1). Jacobi and Laguerre polynomials are instead associated with Beta and Gamma distributions, respectively; see, e.g., [135, 70]. Notice that as for KL-expansions, in order to deal with expansion (3.20) numerically, truncation after a certain number of terms is, in general, required. In this work, we will encounter gPC expansions in subsequent chapters when representing the solution of parametric elliptic boundary value problems.

Chapter 4

Discretisation of elliptic problems with parametric uncertainty

In the present chapter, we focus the attention on the class of elliptic problems under consideration in this work as well as the assumptions used in the remainder of the thesis. We consider a boundary value problem, with an uncertain spatially-varying diffusion coefficient, posed on a polygonal domain in $\mathbb{R}^2$. This coefficient is seen as a second-order random field (see Section 3.2). In particular, we consider the case of a *parametric* random field depending on a countable infinite number of parameters defined as the images of some (independent) random variables. In order for the problem to be well-posed, we make particular assumptions on the random field as well as on the structure of the underlying probability space. Lebesgue-Bochner spaces (or equivalently, tensor products of Hilbert spaces) will be the natural spaces to use in order to derive the weak formulation of the problem.

We consider the numerical approximation of the (weak) solution to our parametric model problem under the general context of spatial discretisations made by means of finite element approximations. Very popular numerical methods to tackle parametric PDE problems are classic Monte Carlo (MC) methods, including, e.g., multi-level Monte Carlo (MLMC) methods (see, e.g., [43, 126, 73]), and stochastic collocation (SC) methods (see, e.g., [5, 103, 102]). These are sampling-based methods that, in particular, require the (a-priori) truncation of input random fields. Furthermore, they can be used also in the case of models where random parameters do not need to take values in bounded domains (e.g., as in the case of lognormal random fields, see Example 3.3.3). Nevertheless, in this thesis, we are not going to consider such class of numerical methods for the discretisation of parametric PDEs. We rather focus on the stochastic Galerkin Finite Element Method (SGFEM); see, e.g., [48, 49, 8, 9]. This method arose by the pioneering con-

tribution [72] which proposed truncated gPC expansions of solutions to parametric PDEs under a Galerkin framework. This has resulted in the development of spectral stochastic finite element methods, where gPC expansions of the solutions (which take into account the stochastic approximation) are combined with a Galerkin projection onto (finite-dimensional) finite element spaces for the spatial approximation (see also [9, 91]).

In what follows, we introduce the model problem, we derive its weak formulation, and state the assumptions needed to ensure the well-posedness in Section 4.1. Then we describe the SGFEM method and provide some details on the implementation aspects in Section 4.2.

4.1 Parametrisation of random inputs

Our starting point is a brief review of the abstract setting of general parametric operator equations. This is the functional analytic setting for the class of elliptic problems with random inputs that deserves our attention.

4.1.1 The abstract setting of parametric operator equations

Throughout, we follow closely the description given in [118]; see also [75].

Let Γ be a compact topological space and $\mathcal{H}$ be a separable Hilbert space over $\mathbb{R}$ equipped with norm $\|\cdot\|_{\mathcal{H}}$. Also, let $\mathcal{L}(\mathcal{H},\mathcal{H}')$ be the space of bounded linear operators from $\mathcal{H}$ to its dual space $\mathcal{H}'$, and $\langle\cdot,\cdot\rangle$ be the duality pairing between $\mathcal{H}'$ and $\mathcal{H}$. Consider the following *parametric operator equation*

$$\mathcal{A}(y)u(y) = f(y) \qquad \forall\, y \in \Gamma, \tag{4.1}$$

where $\mathcal{A} : \Gamma \to \mathcal{L}(\mathcal{H},\mathcal{H}')$ is a continuous map with a bounded inverse $\mathcal{A}^{-1}$ for all $y \in \Gamma$ and $f : \Gamma \to \mathcal{H}'$. The unique solution $u := \mathcal{A}^{-1}f : \Gamma \to \mathcal{H}$ to equation (4.1) is continuous if and only if f is continuous.

In order to derive the weak formulation of (4.1) in the parameter $y \in \Gamma$, we further assume that $\mathcal{A}(y)$ is a symmetric and positive definite operator for all $y \in \Gamma$, and that there exists two positive constants $\mathcal{A}_{\min}, \mathcal{A}_{\max} < \infty$ such that

$$\|\mathcal{A}(y)\|_{\mathcal{L}(\mathcal{H},\mathcal{H}')} \le \mathcal{A}_{\max} \qquad \text{and} \qquad \|\mathcal{A}(y)^{-1}\|_{\mathcal{L}(\mathcal{H}',\mathcal{H})} \le \mathcal{A}_{\min} \qquad \forall\, y \in \Gamma, \tag{4.2}$$

i.e., for any $v \in \mathcal{H}$, the bilinear form $\langle\mathcal{A}(y)v, v\rangle$ is an inner product on $\mathcal{H}$ which induces a norm

equivalent to $\|\cdot\|_{\mathcal{H}}$. Now, suppose that π is a probability measure on the measurable space $(\Gamma, \mathcal{B}(\Gamma))$. In this way, the operator $\mathcal{A} = \mathcal{A}(y)$ depends on a parameter y in a probability space $(\Gamma, \mathcal{B}(\Gamma), \pi)$ while f becomes a random variable on Γ taking values in $\mathcal{H}$. In particular, let us assume the following regularity for the right-hand side

$$f \in L^2_\pi(\Gamma; \mathcal{H}'). \tag{4.3}$$

The weak formulation of problem (4.1) is then defined as follows: find $u \in L^2_\pi(\Gamma; \mathcal{H})$ such that

$$\mathcal{B}(u, v) := \int_\Gamma \langle \mathcal{A}(y)u(y), v(y)\rangle \, d\pi(y) = \int_\Gamma \langle f(y), v(y)\rangle \, d\pi(y) =: \mathcal{F}(v) \qquad \forall\ v \in L^2_\pi(\Gamma; \mathcal{H}). \tag{4.4}$$

Due to assumptions (4.2) and (4.3), both integrals in (4.4) are well defined and the problem above admits a unique solution (see [118, Theorem 2.18]).

4.1.2 Parametric model problem

Hereafter, we refer to the compact topological space Γ as the *parameter space*. Let $D \subset \mathbb{R}^2$ be a bounded domain with a Lipschitz polygonal boundary ∂D. For $f \in L^2(D)$, consider the following homogeneous Dirichlet problem for the parametric steady-state diffusion equation

$$\begin{aligned} -\nabla \cdot (a(\mathbf{x}, \mathbf{y}) \nabla u(\mathbf{x}, \mathbf{y})) &= f(\mathbf{x}) \qquad && \mathbf{x} \in D,\ \mathbf{y} \in \Gamma, \\ u(\mathbf{x}, \mathbf{y}) &= 0 && \mathbf{x} \in \partial D,\ \mathbf{y} \in \Gamma, \end{aligned} \tag{4.5}$$

where $\mathbf{y} \in \Gamma$ are the *parameters*, $\nabla\cdot$ denotes the divergence operator with ∇ denoting the differentiation with respect to spatial variables $\mathbf{x} \in D$, and the diffusion coefficient $a : D \times \Gamma \to \mathbb{R}$ is a second-order random field (see Section 3.2).

Problem (4.5) is an example of a PDE with parametric inputs which perfectly fits into the class of problems whose weak formulations derive from abstract parametric operator equations (4.1) (see Section 4.1.4 below). Here, the parameter $\mathbf{y}$ can be seen as the image in Γ of some random variable $\boldsymbol{Y}$ on a probability space $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$ with known/given probability distribution π. This is indeed one of the assumptions that we are going to make in next section. That is, we suppose to work over the 'image' probability space $(\Gamma, \mathcal{B}(\Gamma), \pi)$, where $\Gamma := \boldsymbol{Y}(\Omega)$, rather than over $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$.

For simplicity, we only consider the case of non-parametric, i.e., *deterministic*, sources $f = f(\mathbf{x})$; in fact, in principle, the source term may also be uncertain, that is, it may depend on $\mathbf{y} \in \Gamma$, thus being a random field. Problem (4.5) is the same parametric equation that is also considered in, e.g., [49, 9, 24, 29, 54, 55, 76]. Note that the solution u to problem (4.5) will be itself a random field.

4.1.3 Main assumptions

Let $(\Gamma, \mathcal{B}(\Gamma), \pi)$ be the underlying probability space for problem (4.5). Let us make the following first assumption.

Assumption 4.1. We assume that

- the parameter domain Γ is the product space

$$\Gamma := \bigotimes_{m=1}^{\infty} \Gamma_m \quad \text{with} \quad \Gamma_m := [-1, 1] \quad \forall\, m \in \mathbb{N}, \tag{4.6}$$

  and $\mathbf{y} \in \Gamma$ are vectors of parameters $y_m \in \Gamma_m$ which are the images of i.i.d. random variables $Y_m : \Omega \to \Gamma_m$ on a probability space $(\Omega, \mathcal{B}(\Omega), \mathbb{P})$, i.e., $\mathbf{y} = (y_m)_{m\in\mathbb{N}} = (Y_m(\omega))_{m\in\mathbb{N}}$, $\omega \in \Omega$. In particular, $\mathcal{B}(\Gamma)$ is the product Borel σ-algebra on Γ (see Section 3.1);

- the probability distribution π is the product probability measure on $(\Gamma, \mathcal{B}(\Gamma))$ given by

$$\pi(\mathbf{y}) := \prod_{m=1}^{\infty} \pi_m(y_m) \qquad \forall\, \mathbf{y} \in \Gamma, \tag{4.7}$$

  where probability distributions π_m are probability measures on $(\Gamma_m, \mathcal{B}(\Gamma_m))$, for all $m \in \mathbb{N}$. Moreover, every measure π_m is *symmetric*, i.e., the probability density $\rho_m : \Gamma_m \to [0, +\infty)$ of Y_m, such that $d\pi_m(y_m) = \rho_m(y_m) dy_m$ (cf. (3.2)), is an even function.

We will refer to $\mathbf{y} \in \Gamma$ as well as the components $y_m \in \Gamma_m$, for all $m \in \mathbb{N}$, as the parameters. We now make the following assumptions on the coefficient $a(\mathbf{x}, \mathbf{y})$ in (4.5).

Assumption 4.2. The diffusion coefficient $a(\mathbf{x}, \mathbf{y})$ is a random field which depends *linearly* on parameters y_m as follows,

$$a(\mathbf{x}, \mathbf{y}) = a_0(\mathbf{x}) + \sum_{m=1}^{\infty} a_m(\mathbf{x})\, y_m, \quad \mathbf{x} \in D,\ \mathbf{y} \in \Gamma, \tag{4.8}$$

where we assume that:

- $a_0(\mathbf{x})$ is uniformly bounded away from zero, i.e., there exist two constants $a_0^{\min}, a_0^{\max} < \infty$ such that

$$0 < a_0^{\min} \le a_0(\mathbf{x}) \le a_0^{\max} \qquad \text{a.e. in } D; \tag{4.9}$$

- spatial functions $(a_m)_{m\in\mathbb{N}}$ satisfy $\|a_m\|_{L^\infty(D)} \ge \|a_{m+1}\|_{L^\infty(D)}$ for all $m \in \mathbb{N}$, and the series in (4.8) converges uniformly in $L^\infty(D)$, with

$$\gamma := \frac{1}{a_0^{\min}} \sum_{m=1}^{\infty} \|a_m\|_{L^\infty(D)} < 1. \tag{4.10}$$

For example, decomposition (4.8) may come from a Karhunen-Loève expansion of a random field $a(\mathbf{x},\mathbf{y})$ with a given covariance function $\mathrm{Cov}[a]$ (cf. (3.13)). Notice that the parameter-free term $a_0(\mathbf{x})$ is equal to the mean value $\mathbb{E}[a]$. In fact, for all $m \in \mathbb{N}$, we have that

$$\int_{\Gamma} y_m \, d\pi(\mathbf{y}) = \left(\prod_{n \neq m}^{\infty} \int_{\Gamma_n} d\pi_n(y_n) \right) \left(\int_{\Gamma_m} y_m \, d\pi_m(y_m) \right) = \int_{\Gamma_m} y_m \rho_m(y_m) dy_m = 0,$$

where the second equality follows since π_m are distribution measures and the third equality follows from the symmetry of intervals Γ_m and since $y_m \rho_m(y_m)$ is an odd function. In addition, for all $\mathbf{y} \in \Gamma$, we have

$$|\, a(\mathbf{x},\mathbf{y}) - a_0(\mathbf{x}) \,| \leq \sum_{m=1}^{\infty} |a_m(\mathbf{x}) y_m| \overset{(4.6)}{\leq} \sum_{m=1}^{\infty} |a_m(\mathbf{x})| \leq \sum_{m=1}^{\infty} \| a_m(\mathbf{x}) \|_{L^\infty(D)} \overset{(4.10)}{<} a_0^{\min},$$

and from (4.9) we see that the random field $a(\mathbf{x},\mathbf{y})$ in (4.8) is positive and bounded from above by $a_0^{\min} + a_0^{\max}$ a.e. in D.

Remark 4.1.1 (Finite-dimensional noise assumption)**.** *The main theoretical and numerical challenge in dealing with PDE problems depending on a countable infinite number of parameters is that, in general, one does not known a priori which and how many parameters should be incorporated in the discretisation of the model. In other words, it is difficult to decide after how many terms the representation of random inputs should be truncated. However, in many practical settings, realisations of the input random field may be slowly varying in space. When this happens, only few terms in the series are effectively needed to capture accurately the features of the random field. In this case, expansions such as* (3.13) *or* (4.8) *can be truncated after a certain number of terms (cf.* (3.14)*). In literature, this is also known as finite-dimensional noise assumption; see, e.g., [8, 68, 5, 79]. In our current setting, we keep assumption* (4.8) *since we will show later that the approximation of problem* (4.5) *via SGFEM naturally leads to truncated series in the expansion of the random field (see Proposition 4.1).*

It is worth recalling that we do not loose generality in assuming that bounded parameters y_m take values in $\Gamma_m = [-1,1]$ ($m \in \mathbb{N}$) as in Assumption 4.1, since this can be always ensured by rescaling appropriately the spatial functions a_m in (4.8) (see [118, Lemma 2.20]).

4.1.4 Weak formulation

Let $H_0^1(D)$ be the Sobolev space of functions in $H^1(D)$ vanishing on ∂D in the sense of traces and $H^{-1}(D)$ be its dual space. Furthermore, let $\langle \cdot, \cdot \rangle$ be the duality pairing between $H^{-1}(D)$ and $H_0^1(D)$.

For all $\mathbf{y} \in \Gamma$, let $f(\mathbf{y}) \in H^{-1}(D)$ be defined by $\langle f(\mathbf{y}), w \rangle := \int_D f(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}$ for all $w \in H_0^1(D)$. Fur-

thermore, for all $\mathbf{y} \in \Gamma$, we define the following symmetric operator $A(\mathbf{y}) : H_0^1(D) \to H^{-1}(D)$,

$$\langle A(\mathbf{y})v, w\rangle := \int_D a(\mathbf{x},\mathbf{y})\nabla v(\mathbf{x})\cdot\nabla w(\mathbf{x})d\mathbf{x} = \langle A_0 v, w\rangle + \sum_{m=1}^{\infty} y_m\langle A_m v, w\rangle \quad \forall\ v, w \in H_0^1(D), \tag{4.11}$$

where $A_m : H_0^1(D) \to H^{-1}(D)$ are the symmetric operators defined by

$$\langle A_m v, w\rangle := \int_D a_m(\mathbf{x})\nabla v(\mathbf{x})\cdot\nabla w(\mathbf{x})d\mathbf{x} \qquad \forall\ v, w \in H_0^1(D),\ m \in \mathbb{N}_0. \tag{4.12}$$

Decomposition (4.11) follows the structure of the random field (4.8) and note that operator $A(\mathbf{y})$ in (4.11) is the one associated with problem (4.5) for all $\mathbf{y} \in \Gamma$. Assumption (4.9) on the mean field a_0 of the random field implies that the bilinear form $\langle A_0\cdot,\cdot\rangle$ is both continuous and coercive, i.e.,

$$|\langle A_0 v, w\rangle| \le a_0^{\max}\|v\|_{H_0^1(D)}\|w\|_{H_0^1(D)} \qquad \forall\ v, w \in H_0^1(D), \tag{4.13}$$

$$\langle A_0 v, v\rangle \ge a_0^{\min}\|v\|^2_{H_0^1(D)} \qquad \forall\ v \in H_0^1(D). \tag{4.14}$$

Also, both (4.9) and (4.10) imply that the operator $A(\mathbf{y})$, as well as its inverse $A(\mathbf{y})^{-1}$, are bounded for all $\mathbf{y} \in \Gamma$,

$$\sup_{\mathbf{y}\in\Gamma}\|A(\mathbf{y})\|_{\mathcal{L}(H_0^1(D),H^{-1}(D))} \le A_{\max} \qquad \text{and} \qquad \sup_{\mathbf{y}\in\Gamma}\|A(\mathbf{y})^{-1}\|_{\mathcal{L}(H^{-1}(D),H_0^1(D))} \le A_{\min}^{-1}, \tag{4.15}$$

where the positive positive constants $A_{\min}$ and $A_{\max}$ are given by

$$A_{\max} := a_0^{\max}(1+\gamma) \qquad \text{and} \qquad A_{\min} := a_0^{\min}(1-\gamma), \tag{4.16}$$

respectively (see [118, Proposition 2.22]). Furthermore, (4.10) and (4.13) ensure that the series in (4.11) converges in $\mathcal{L}(H_0^1(D), H^{-1}(D))$ uniformly in $\mathbf{y}$. In particular, $A(\mathbf{y})$ depends continuously on $\mathbf{y} \in \Gamma$ (see [118, Lemma 2.21]).

Now, let us consider the Lebesgue-Bochner space $L^2_\pi(\Gamma; H_0^1(D))$. The weak formulation of problem (4.5) reads as: find $u \in L^2_\pi(\Gamma; H_0^1(D))$ such that (cf. (4.4))

$$B(u,v) := \int_\Gamma \langle A(\mathbf{y})u(\mathbf{y}), v(\mathbf{y})\rangle d\pi(\mathbf{y}) = \int_\Gamma \langle f(\mathbf{y}), v(\mathbf{y})\rangle d\pi(\mathbf{y}) =: F(v) \qquad \forall\ v \in L^2_\pi(\Gamma; H_0^1(D)), \tag{4.17}$$

where, due to (4.11) and (4.12), the symmetric bilinear form B can be written as

$$B(u,v) = B_0(u,v) + \sum_{m=1}^{\infty} B_m(u,v) \qquad \forall\ u, v \in L^2_\pi(\Gamma; H_0^1(D)), \tag{4.18}$$

with component bilinear forms B_0 and B_m given by

$$B_0(u,v) := \int_\Gamma \langle A_0 u(\mathbf{y}), v(\mathbf{y})\rangle\, d\pi(\mathbf{y}) \qquad \forall\ u, v \in L^2_\pi(\Gamma; H_0^1(D)), \tag{4.19}$$

$$B_m(u,v) := \int_\Gamma y_m\langle A_m u(\mathbf{y}), v(\mathbf{y})\rangle\, d\pi(\mathbf{y}) \qquad \forall\ u, v \in L^2_\pi(\Gamma; H_0^1(D)),\ \forall\ m \in \mathbb{N}. \tag{4.20}$$

Note that $F \in L^2_\pi(\Gamma; H^{-1}(D))$ and inequalities (4.15) imply that $B(\cdot,\cdot)$ is both continuous and coercive with $A_{\max}$ and $A_{\min}$ defined in (4.16) being the continuity and coercivity constants, respectively. Therefore, the existence of the unique solution $u \in V$ satisfying the weak problem (4.17) is guaranteed by the Lax-Milgram lemma (see, e.g., [35, Theorem 2.7.7]).

To conclude, observe that on the one hand, the bilinear form $B(\cdot,\cdot)$ defines an inner product in $L^2_\pi(\Gamma; H^1_0(D))$ which induces a norm

$$\|v\|_B := B(v,v)^{1/2} \qquad \forall\, v \in L^2_\pi(\Gamma; H^1_0(D)),$$

called the *energy norm*, that is equivalent to $\|\cdot\|_{L^2_\pi(\Gamma;H^1_0(D))}$ (see, e.g., [75, Lemma 1.3]). On the other hand, (4.13) and (4.14) imply that also $B_0(\cdot,\cdot)$ defines an inner product in $L^2_\pi(\Gamma; H^1_0(D))$ which induces the norm $\|\cdot\|_{B_0} := B_0(\cdot,\cdot)^{1/2}$ equivalent to $\|\cdot\|_{L^2_\pi(\Gamma;H^1_0(D))}$. Therefore, we have the following equivalence of norms,

$$\lambda \|v\|_B^2 \le \|v\|_{B_0}^2 \le \Lambda \|v\|_B^2 \qquad \forall\, v \in L^2_\pi(\Gamma; H^1_0(D)), \tag{4.21}$$

where $\lambda < 1 < \Lambda$ are given by

$$\lambda := a_0^{\min} A_{\max}^{-1} \qquad \text{and} \qquad \Lambda := a_0^{\max} A_{\min}^{-1}. \tag{4.22}$$

4.2 Stochastic Galerkin Finite Element Method

In this section, we describe the approximation of the solution to weak formulation (4.17) using the stochastic Galerkin Finite Element Method (SGFEM); see, e.g., [72, 48, 49, 8, 9, 91, 79].

First of all, notice that the weak solution $u \in L^2_\pi(\Gamma; H^1_0(D))$ to the weak formulation (4.17) can be seen as a function in the tensor product space $L^2_\pi(\Gamma) \otimes H^1_0(D)$ due to isomorphism (2.6). Therefore, in the remainder of the thesis, we choose

$$V := L^2_\pi(\Gamma) \otimes H^1_0(D) \tag{4.23}$$

as both our trial and test space in (4.17). As in non-parametric (i.e., deterministic) FEM, discretisation via SGFEM aims at computing a discrete approximation of a weak solution u via Galerkin projection onto a suitable finite-dimensional subspace of V. Such subspace can be defined by maintaining the same tensor product structure of V, i.e., it can be given by the tensor product of independently constructed finite-dimensional subspaces of $H^1_0(D)$ and $L^2_\pi(\Gamma)$. In particular, the method considers a finite element space associated with the spatial discretisation of D and a

finite-dimensional space of multivariate polynomials in the parameters for the stochastic approximation.

The construction of the finite-dimensional subspace of $L^2_\pi(\Gamma)$ deserves particular attention. We describe such construction in the following section.

4.2.1 Orthogonal polynomials in the parameter space

For each $m \in \mathbb{N}$, let $\{p_k^m\}_{k\in\mathbb{N}_0}$ be the set of univariate polynomials of degree k, with $p_0^m := 1$, which forms an orthonormal basis of $L^2_{\pi_m}(\Gamma_m)$ with respect to the inner product $(\cdot,\cdot)_{\pi_m}$. If π_m has finite *moments*, i.e.,

$$\int_{\Gamma_m} y_m^n \, d\pi_m(y_m) < \infty \qquad \forall\, n \in \mathbb{N}_0, \tag{4.24}$$

these polynomials satisfy the well-known *three-term recurrence* (see, e.g., [70]),

$$\beta_{k+1}^m \, p_{k+1}^m(y_m) = (y_m - \alpha_k^m)\, p_k^m(y_m) - \beta_k^m \, p_{k-1}^m(y_m), \qquad y_m \in \Gamma_m,\ k \in \mathbb{N}_0, \tag{4.25}$$

where $p_{-1}^m := 0$ for all $m \in \mathbb{N}$ and

$$\alpha_k^m := (y_m\, p_k^m, p_k^m)_{\pi_m} \qquad \text{and} \qquad \beta_k^m := c_{k-1}^m / c_k^m,$$

with c_k^m denoting the leading coefficient of p_k^m and $\beta_0^m := 1$. Notice that under Assumption 4.1, we have that $\alpha_k^m = 0$ in (4.25) due to the symmetry of Γ_m and the symmetry of measures π_m, for all $m \in \mathbb{N}$. Recurrence (4.25) can be derived from the Gram-Schmidt orthogonalisation of monomials $(y_m^n)_{n\in\mathbb{N}_0}$. For a proof of orthonormality of a set of (univariate) polynomials $\{p_k^m\}_{k\in\mathbb{N}_0}$ satisfying (4.25), see, e.g., [118, Lemma 2.14]; the completeness, on the other hand, is a classic result due to Riesz based on the assumption that measure π_m is uniquely 'determinated' by its moments in (4.24) (see, e.g., [23, Theorem 2.1] and [63, Section 3.1]).

We report some examples of sets of orthonormal polynomials satisfying the three-term recurrence (4.25). For additional examples of families of orthonormal polynomials we refer to, e.g., [135, 70].

Example 4.2.1 (Legendre polynomials)**.** *For all $m \in \mathbb{N}$, let $\Gamma_m = [-1,1]$ and π_m be the uniform distribution measure on $(\Gamma_m, \mathcal{B}(\Gamma_m))$, i.e., $d\pi_m(y_m) = \rho_m(y_m)dy_m$ with $\rho_m(y_m) = 1/2$ for all $y_m \in \Gamma_m$. Then, the orthonormal basis of polynomials of $L^2_\pi(\Gamma_m)$ consists of scaled Legendre polynomials, defined*

by Rodrigues' formula

$$L_k^m(y_m) := \frac{\sqrt{2k+1}}{2^k k!}\frac{d^k}{dy_m^k}(y_m^2-1), \qquad y_m \in \Gamma_m,\ k \in \mathbb{N}_0,$$

which satisfy the following three-term recurrence

$$\beta_{k+1}^m L_{k+1}^m(y_m) = y_m L_k^m(y_m) - \beta_k^m L_{k-1}^m(y_m), \quad k \in \mathbb{N}_0, \tag{4.26}$$

with $L_{-1}^m := 0$, $L_0^m := 1$, *and* $\beta_0^m := 1$ *and* $\beta_k^m = k/(\sqrt{2k+1}\sqrt{2k-1})$ *for all* $k \in \mathbb{N}$.

Example 4.2.2 (Hermite polynomials)**.** *For all* $m \in \mathbb{N}$, *let* $\Gamma_m := \mathbb{R}$ *and* π_m *be the standard normal distribution measure on* $(\Gamma_m, \mathcal{B}(\Gamma_m))$, *i.e.,* $d\pi_m(y_m) = \rho_m(y_m)dy_m$ *with*

$$\rho_m(y_m) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y_m^2}{2}\right), \qquad y_m \in \Gamma_m.$$

Then, the orthonormal basis of polynomials of $L^2_\pi(\Gamma_m)$ *consists of normalised Hermite polynomials, defined by Rodrigues' formula*

$$H_k^m(y_m) := (-1)^k \frac{1}{\sqrt{k!}}\exp\left(\frac{y_m^2}{2}\right)\frac{d^k}{dy_m^k}\exp\left(-\frac{y_m^2}{2}\right), \qquad y_m \in \Gamma_m,\ k \in \mathbb{N}_0,$$

which satisfy the following three-term recurrence

$$\beta_{k+1}^m H_{k+1}^m(y_m) = y_m H_k^m(y_m) - \beta_k^m H_{k-1}^m(y_m), \quad k \in \mathbb{N}_0,$$

with $H_{-1}^m := 0$, $H_0^m := 1$, *and* $\beta_0^m := 1$, $\beta_k^m = \sqrt{k}$ *for all* $k \in \mathbb{N}$.

Example 4.2.3 (Rys polynomials)**.** *For all* $m \in \mathbb{N}$, *let* $\Gamma_m = [-b, b]$ *with* $b \in \mathbb{R}$, *and let* π_m *the 'truncated' Gaussian distribution measure on* $(\Gamma_m, \mathcal{B}(\Gamma_m))$, *i.e.,* $d\pi_m(y_m) = \rho_m(y_m)dy_m$ *with*

$$\rho_m(y_m) = \left(2\Phi\left(\frac{b}{s}\right)-1\right)^{-1}\frac{1}{\sqrt{2\pi s^2}}\exp\left(-\frac{y_m^2}{2s^2}\right)\chi_{[-b,b]}(y_m), \qquad y_m \in \Gamma_m, \tag{4.27}$$

where $\Phi(\cdot)$ *denotes the Gaussian cumulative distribution function defined as*

$$\Phi(x) := \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\exp\left(-\frac{t^2}{2}\right)dt \qquad \forall\, x \in \mathbb{R},$$

$s \in \mathbb{R}$, *and* $\chi_{[-b,b]}$ *is the characteristic function of the interval* $[-b, b]$. *The function defined by* (4.27) *is the probability density function truncated on the interval* $[-b, b]$ *of a mean zero Gaussian random variable with standard deviation* s. *Hence,* π_m *is a particular case of the measure in Example 4.2.2. The associated orthonormal polynomials in* $L^2_{\pi_m}(\Gamma_m)$ *which satisfy the three-term recurrence* (4.25) *are typically referred to as* Rys *polynomials; see, e.g., [83, 116].*

Now, let $\mathbb{N}_0^{\mathbb{N}} := \{\nu = (\nu_m)_{m\in\mathbb{N}} : \nu_m \in \mathbb{N}_0\}$ be the set of multi-indices. In order to construct a basis

for the space $L^2_\pi(\Gamma)$, we define the following set of *finitely supported* multi-indices

$$\mathcal{J} := \left\{\nu \in \mathbb{N}_0^{\mathbb{N}} \;:\; \#\mathrm{supp}(\nu) < \infty\right\}, \tag{4.28}$$

where $\mathrm{supp}(\nu) := \{m \in \mathbb{N} : \nu_m > 0\}$ is the *support* of ν and # denotes the cardinality. Note that the set $\mathcal{J}$ is countable since it can be understood as a countable union of countable sets. The set $\mathcal{J}$ and all its subsets are called *index sets*, and we refer to elements $\nu \in \mathcal{J}$ simply as *indices*. For $\mathbf{y} \in \Gamma$, we define the following polynomials

$$P_\nu(\mathbf{y}) := \prod_{m=1}^{\infty} p^m_{\nu_m}(y_m) = \prod_{m\in\mathrm{supp}(\nu)} p^m_{\nu_m}(y_m) \qquad \forall\, \nu \in \mathcal{J}, \tag{4.29}$$

where $\{p^m_{\nu_m}\}$ is the set of polynomials of degree ν_m, with $p^m_0 := 1$, forming a basis of $L^2_{\pi_m}(\Gamma_m)$ for all $m \in \mathbb{N}$. Then, the set $\{P_\nu(\mathbf{y})\}_{\nu\in\mathcal{J}}$ of polynomials P_ν defined in (4.29), for all $\nu \in \mathcal{J}$, is an orthonormal basis of $L^2_\pi(\Gamma)$ with respect to inner product $(\cdot,\cdot)_\pi$ (see, e.g., [118, Theorem 2.12]). Note that due to the tensor product structure of the space V, each function $v \in V$ can be written as a PC expansion in the orthonormal polynomials P_ν, i.e.,

$$v(\mathbf{x},\mathbf{y}) = \sum_{\nu\in\mathcal{J}} w_\nu(\mathbf{x}) P_\nu(\mathbf{y}) \quad \text{with unique coefficients } w_\nu \in H^1_0(D). \tag{4.30}$$

For a given finite index set $\mathcal{P} \subset \mathcal{J}$ of cardinality $\#\mathcal{P} < \infty$, we denote by $\mathcal{P}_\mathcal{P}$ the finite-dimensional subspace of $L^2_\pi(\Gamma)$ of polynomials associated with $\mathcal{P}$, i.e.,

$$\mathcal{P}_\mathcal{P} := \mathrm{span}\left\{P_\nu(\mathbf{y}) \;:\; \nu \in \mathcal{P},\ \mathbf{y} \in \Gamma\right\} = \bigoplus_{\nu\in\mathcal{P}} \mathcal{P}_\nu, \tag{4.31}$$

where $\mathcal{P}_\nu := \mathrm{span}\{P_\nu\}$ and P_ν is defined in (4.29) for each $\nu \in \mathcal{P}$. Throughout, we will assume that $\mathcal{P}_\mathcal{P}$ in (4.31) always contains the *zero index* $\nu = \mathbf{0} := (0,0,\dots)$ so that $\mathcal{P}_\mathcal{P}$ also contains constant functions.

Example 4.2.4. *Let $\mathcal{P} \subset \mathcal{J}$ be a finite subset consisting of the following four indices:*

$$\mathcal{P} := \left\{\nu^{(1)} = \mathbf{0},\ \nu^{(2)} = (1,0,0,\dots),\ \nu^{(3)} = (0,1,0,\dots),\ \nu^{(4)} = (2,1,0,\dots)\right\}.$$

We have $\mathrm{supp}(\nu^{(1)}) = \emptyset$, $\mathrm{supp}(\nu^{(2)}) = \{1\}$, $\mathrm{supp}(\nu^{(3)}) = \{2\}$, *and* $\mathrm{supp}(\nu^{(4)}) = \{1,2\}$. *From* (4.29) *and* (4.31), *it follows that the basis of the associated space $\mathcal{P}_\mathcal{P}$ is formed of the following four polynomials*

$$P_{\nu^{(1)}}(\mathbf{y}) = 1, \quad P_{\nu^{(2)}}(\mathbf{y}) = p^1_1(y_1), \quad P_{\nu^{(3)}}(\mathbf{y}) = p^2_1(y_2), \quad P_{\nu^{(4)}}(\mathbf{y}) = p^1_2(y_1)p^2_1(y_2), \qquad \forall\, \mathbf{y} \in \Gamma.$$

For example, in the case of uniformly distributed parameters on $\Gamma_m = [-1,1]$, the orthonormal basis $\{p^m_k\}_{k\in\mathbb{N}_0}$ of $L^2_{\pi_m}(\Gamma_m)$ consists of Legendre polynomials satisfying the three-term recurrence (4.26), *thus*

we have

$$P_{\nu^{(1)}} = 1, \quad P_{\nu^{(2)}} = y_1\sqrt{3}, \quad P_{\nu^{(3)}} = y_2\sqrt{3}, \quad P_{\nu^{(4)}} = \left(\sqrt{5}(3y_1^2-1)/2\right)\left(y_2\sqrt{3}\right).$$

Example 4.2.4 shows that a finite index set $\mathcal{P} \subset \mathcal{J}$ determines both the 'active' parameters y_m of $\mathcal{P}$, i.e., those parameters $y_m \in \Gamma_m$ for which there exists $\nu \in \mathcal{P}$ with nonzero ν_m (y_1 and y_2 in the example) as well as the associated polynomial degrees in these 'active' parameters. This observations lead to the following definition.

Definition 4.1. *Let $\mathcal{P}$ be a finite subset of $\mathcal{J}$. The* support of $\mathcal{P}$ *is defined as*

$$\operatorname{supp}(\mathcal{P}) := \bigcup_{\nu\in\mathcal{P}} \operatorname{supp}(\nu). \tag{4.32}$$

A parameter $y_m \in \Gamma_m$ is active *(in $\mathcal{P}$) if $m \in \operatorname{supp}(\mathcal{P})$. Furthermore, we denote by $M_{\mathcal{P}} := \#\operatorname{supp}(\mathcal{P})$ the number of active parameters of $\mathcal{P}$.*

With reference to previous Example 4.2.4, we have $\operatorname{supp}(\mathcal{P}) = \{1,2\}$, $M_{\mathcal{P}} = 2$, and y_1 and y_2 as active parameters in $\mathcal{P}$.

Example 4.2.5 (Complete and tensor product polynomials)**.** *Let $M \in \mathbb{N}$ and $n \in \mathbb{N}_0$. The space of complete polynomials $\mathcal{P}_{\mathcal{P}(M,n)}$ is the space of polynomials of total degree less than or equal to n in the first M parameters y_m, i.e., the associated index set $\mathcal{P}(M,n)$ is defined as*

$$\mathcal{P}(M,n) := \left\{ \nu \in \mathbb{N}_0^{\mathbb{N}} \ : \ \operatorname{supp}(\nu) \subseteq \{1,\dots,M\}, \ \sum_{m=1}^{M} \nu_m \le n \right\}. \tag{4.33}$$

The space of tensor product polynomials $\mathcal{P}_{\widetilde{\mathcal{P}}(M,n)}$ is the space of polynomials of degree n in each of the parameters y_m for $m = 1,\dots,M$, i.e., the associated index set $\widetilde{\mathcal{P}}(M,n)$ is defined as

$$\widetilde{\mathcal{P}}(M,n) := \left\{ \nu \in \mathbb{N}_0^{\mathbb{N}} \ : \ \operatorname{supp}(\nu) \subseteq \{1,\dots,M\}, \ \nu_m \le n \ \text{ for } m = 1,\dots,M \right\}. \tag{4.34}$$

Notice that $\mathcal{P}(M,n) \subseteq \widetilde{\mathcal{P}}(M,n)$ and $\operatorname{supp}(\mathcal{P}) = \operatorname{supp}(\widetilde{\mathcal{P}}) = \{1,\dots,M\}$, i.e., only the first M parameters are active. For example, for $M = n = 2$,

$$\mathcal{P}(2,2) = \begin{Bmatrix} (0,0), & (1,0), & (2,0) \\ (0,1), & (1,1) & \\ (0,2) & & \end{Bmatrix} \quad \text{and} \quad \widetilde{\mathcal{P}}(2,2) = \begin{Bmatrix} (0,0), & (1,0), & (2,0) \\ (0,1), & (1,1), & (2,1) \\ (0,2), & (1,2), & (2,2) \end{Bmatrix}.$$

The dimensions of these spaces are $\#\mathcal{P}(M,n) = (M+n)!/(M!\,n!)$ and $\#\widetilde{\mathcal{P}}(M,n) = (n+1)^M$, respectively.

4.2.2 Discrete weak formulation

For the construction of a finite-dimensional subspace of $H_0^1(D)$ for the spatial discretisation of weak problem (4.17), the SGFEM considers a finite element space. For simplicity, here, we focus on the first-order finite element space $X := \mathcal{S}_0^1(\mathcal{T})$ of piecewise linear functions on a conforming triangulation $\mathcal{T}$ of D (see (2.12)).

With both the finite element space $X \subset H_0^1(D)$ and the polynomial space $\mathcal{P}_\mathcal{P} \subset L_\pi^2(\Gamma)$ for a given finite index set $\mathcal{P} \subset \mathcal{J}$ (see (4.31)), we define the finite-dimensional subspace $V_{X\mathcal{P}}$ of V as

$$V_{X\mathcal{P}} := X \otimes \mathcal{P}_\mathcal{P} \subset V. \tag{4.35}$$

Then, the associated discrete weak formulation of problem (4.17) reads: find $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ such that

$$B(u_{X\mathcal{P}}, v) = F(v) \qquad \forall\, v \in V_{X\mathcal{P}}. \tag{4.36}$$

As for (4.17), the uniqueness of solution $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ to problem (4.36) follows from the Lax-Milgram lemma. Moreover, the following best approximation property holds (see, e.g., [88, Theorem 9.51]),

$$\|u - u_{X\mathcal{P}}\|_B = \inf_{v \in V_{X\mathcal{P}}} \|u - v\|_B. \tag{4.37}$$

In addition, due to the tensor product structure of $V_{X\mathcal{P}}$, the Galerkin solution $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ can be written as

$$u_{X\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \sum_{\nu \in \mathcal{P}} \phi_\nu(\mathbf{x}) P_\nu(\mathbf{y}), \qquad \mathbf{x} \in D,\ \mathbf{y} \in \Gamma, \tag{4.38}$$

with unique PC coefficients $\phi_\nu \in X$ that are finite element functions for all $\nu \in \mathcal{P}$. The standard SGFEM approximation (4.38) satisfying problem (4.36) on the tensor product space $V_{X\mathcal{P}}$ in (4.35) can be also referred to as *single-level* approximation. The name refers to the fact that each PC coefficient ϕ_ν ($\nu \in \mathcal{P}$) is defined on the *same* finite element space X (see Remark 5.2.3).

It is worth noticing that although (4.36) represents a discrete problem on the finite-dimensional subspace $V_{X\mathcal{P}} \subset V$, the bilinear form $B(\cdot,\cdot)$ on the left-hand side of (4.36) is given by (4.18) which contains a series representation. Therefore, for the approximation $u_{X\mathcal{P}}$ to be computable it is necessary that only a finite number of terms are nonzero in $B(u_{X\mathcal{P}}, v)$ in (4.36). The following results shows that this is indeed guaranteed by the fact that $\mathcal{P}$ is a finite set of indices.

Proposition 4.1. *Under Assumption 4.1, let $\mathcal{P}$ be a finite subset of $\mathcal{J}$ with $M_\mathcal{P} = \#\mathrm{supp}(\mathcal{P}) < \infty$ active parameters. Then, the series in the bilinear form $B(\cdot,\cdot)$ on the left-hand side of* (4.36) *only involves the*

$M_{\mathcal{P}}$ *terms indexed by* $m \in \operatorname{supp}(\mathcal{P})$, *i.e.*,

$$B(u_{X\mathcal{P}}, v) \;=\; B_0(u_{X\mathcal{P}}, v) \;+\; \sum_{m \in \operatorname{supp}(\mathcal{P})} B_m(u_{X\mathcal{P}}, v) \qquad \forall\; v \in V_{X\mathcal{P}}\,, \tag{4.39}$$

where B_0 *and* B_m *are given by* (4.19) *and* (4.20), *respectively.*

Proof. Let $u_{X\mathcal{P}}$ be as in (4.38) and, analogously, let $v(\mathbf{x},\mathbf{y}) = \sum_{\mu\in\mathcal{P}} \psi_\mu(\mathbf{x}) P_\mu(\mathbf{y}) \in V_{X\mathcal{P}}$ with finite element coefficients $\psi_\mu \in X$. Due to (4.18)–(4.20) and the orthonormality of polynomials $P_\nu \in \mathcal{P}_{\mathcal{P}}$, we have that

$$B(u_{X\mathcal{P}}, v) = \sum_{\nu\in\mathcal{P}} \langle A_0\phi_\nu(\mathbf{x}), \psi_\nu(\mathbf{x})\rangle \;+\; \sum_{m=1}^{\infty} \left(\sum_{\nu,\mu\in\mathcal{P}} \langle A_m\phi_\nu(\mathbf{x}), \psi_\mu(\mathbf{x})\rangle \, (y_m P_\nu(\mathbf{y}), P_\mu(\mathbf{y}))_\pi \right).$$

Expanding the inner product $(\cdot,\cdot)_\pi$ in the above equation, we obtain

$$(y_m P_\nu, P_\mu)_\pi \;=\; \left(\prod_{s\in\mathbb{N}\setminus\{m\}} \int_{\Gamma_s} p^s_{\nu_s}(y_s) p^s_{\mu_s}(y_s) d\pi_s(y_s) \right) \left(\int_{\Gamma_m} y_m \, p^m_{\nu_m}(y_m) p^m_{\mu_m}(y_m) d\pi_m(y_m) \right). \tag{4.40}$$

Since $\mathcal{P}$ has $M_{\mathcal{P}} = \#\operatorname{supp}(\mathcal{P})$ active parameters, then $\nu_m = \mu_m = 0$ and $p^m_{\nu_m}(y_m) = p^m_{\mu_m}(y_m) = 1$ when $m \notin \operatorname{supp}(\mathcal{P})$. Therefore, for all $m \notin \operatorname{supp}(\mathcal{P})$, the second term in brackets on the right-hand side of (4.40) is zero,

$$\int_{\Gamma_m} y_m d\pi_m(y_m) = 0\,, \tag{4.41}$$

since π_m are symmetric measures and $\Gamma_m = [-1,1]$ for all $m \in \mathbb{N}$. □

Proposition 4.1 shows that $(y_m P_\nu, P_\mu)_\pi = 0$ for all $m \notin \operatorname{supp}(\mathcal{P})$, $\nu,\mu \in \mathcal{P}$, under the symmetry of measures π_m and the symmetry of domains Γ_m. Generally, this also holds if parameters y_m are the images of mean zero random variables Y_m for all $m \in \mathbb{N}$. In this case, for all $m \notin \operatorname{supp}(\mathcal{P})$, we have that $\int_{\Gamma_m} y_m d\pi_m(y_m) = \mathbb{E}[Y_m] = 0$, i.e., (4.41) holds.

Corollary 4.1. *Let* $\mathcal{P} \subset \mathcal{I}$ *be a finite subset with* $M_{\mathcal{P}} = \#\operatorname{supp}(\mathcal{P}) < \infty$ *active parameters* y_m *which are the images of mean zero random variables for all* $m \in \mathbb{N}$. *Then* (4.39) *holds.*

It is clear that whenever $\operatorname{supp}(\mathcal{P}) = \{1, 2, \ldots, M\}$, i.e., only the first $M = M_{\mathcal{P}} \in \mathbb{N}$ parameters are active, then (4.39) reads as

$$B(u_{X\mathcal{P}}, v) \;=\; B_0(u_{X\mathcal{P}}, v) \;+\; \sum_{m=1}^{M} B_m(u_{X\mathcal{P}}, v) \qquad \forall\; v \in V_{X\mathcal{P}}\,. \tag{4.42}$$

Hereafter, if this property holds, we say that the index set $\mathcal{P}$ is *ordered*. Furthermore, notice that when a parameter y_m is active, i.e., for $m \in \operatorname{supp}(\mathcal{P})$, we also implicitly have the associated coeffi-

cient a_m 'active' in random field (4.8) as well as in the Galerkin discretisation, since a_m is incorporated in the nonzero term $B_m(u_{X\mathcal{P}}, v)$ (see (4.20)).

4.2.3 Stochastic Galerkin linear system

Let $\{\varphi_j\}_{j=1}^{N_X}$ be the set of basis functions of X, with $N_X := \#\mathcal{N}^\circ(\mathcal{T})$ denoting the number of interior vertices of the underlying triangulation $\mathcal{T}$ of the domain D. Let $\mathcal{P} \subset \mathcal{J}$ be a ordered finite index set of cardinality $N_\mathcal{P} := \#\mathcal{P}$ with the first $M_\mathcal{P} = \#\mathrm{supp}(\mathcal{P})$ parameters active. In what follows, we identify the indices $\nu \in \mathcal{P}$ and the associated polynomials $P_\nu \in \mathcal{P}_\mathcal{P}$ by using an integer $s \in \mathbb{N}$. In particular, we denote by $P_s(\mathbf{y})$ the polynomial $P_{\nu^{(s)}}(\mathbf{y}) \in \mathcal{P}_\mathcal{P}$ associated with the s-th index $\nu^{(s)} \in \mathcal{P}$, for all $s = 1, \ldots, N_\mathcal{P}$, and we further assume that the first index $\nu^{(1)} \in \mathcal{P}$ is the zero index, i.e., $\nu^{(1)} = \mathbf{0}$, such that $P_1(\mathbf{y}) = 1$ for all $\mathbf{y} \in \Gamma$.

Let us rewrite the polynomial chaos expansion (4.38) using this notation:

$$u_{X\mathcal{P}}(\mathbf{x}, \mathbf{y}) \;=\; \sum_{s=1}^{N_\mathcal{P}} \phi_s(\mathbf{x}) P_s(\mathbf{y}) \;=\; \sum_{s=1}^{N_\mathcal{P}} \sum_{j=1}^{N_X} u_{sj} \varphi_j(\mathbf{x}) P_s(\mathbf{y}), \qquad \mathbf{x} \in D,\ \mathbf{y} \in \Gamma, \tag{4.43}$$

where $\{u_{sj}\}$, $s = 1, \ldots, N_\mathcal{P}$ and $j = 1, \ldots, N_X$, are real coefficients which have to be computed as explained below. Let $N_{X\mathcal{P}} := N_X N_\mathcal{P} = \dim(X \otimes \mathcal{P}_\mathcal{P})$ denoting the dimension of the discrete space $V_{X\mathcal{P}}$. With $u_{X\mathcal{P}}$ given by (4.43), and choosing a test function $v_{X\mathcal{P}} = \varphi_i(\mathbf{x}) P_t(\mathbf{y}) \in V_{X\mathcal{P}}$ for $i = 1, \ldots, N_X$ and $t = 1, \ldots, N_\mathcal{P}$, the discrete weak formulation (4.36) yields the following linear system

$$\boldsymbol{B}\boldsymbol{u} = \boldsymbol{F} \qquad \text{with} \qquad \boldsymbol{B} = \begin{pmatrix} B_{11} & B_{12} & \ldots & B_{1N_\mathcal{P}} \\ B_{21} & B_{22} & \ldots & B_{2N_\mathcal{P}} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\mathcal{P}1} & B_{N_\mathcal{P}2} & \ldots & B_{N_\mathcal{P}N_\mathcal{P}} \end{pmatrix}, \quad \boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_{N_\mathcal{P}} \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \vdots \\ \boldsymbol{F}_{N_\mathcal{P}} \end{pmatrix}. \tag{4.44}$$

Here, $\boldsymbol{u} \in \mathbb{R}^{N_{X\mathcal{P}}}$ is the solution vector whose solution blocks $\boldsymbol{u}_s \in \mathbb{R}^{N_X}$ contain the N_X coefficients associated with indices $\nu^{(s)} \in \mathcal{P}$, i.e.,

$$\boldsymbol{u}_s \;=\; (u_{s1}, \ldots, u_{sN_X})^T, \qquad s = 1, \ldots, N_\mathcal{P}.$$

Each block $B_{ts} \in \mathbb{R}^{N_X \times N_X}$ of matrix $\boldsymbol{B} \in \mathbb{R}^{N_{X\mathcal{P}} \times N_{X\mathcal{P}}}$ in (4.44) has the form

$$B_{ts} \;=\; (P_s, P_t)_\pi K_0 \;+\; \sum_{m=1}^{M_\mathcal{P}} (y_m P_s, P_t)_\pi K_m, \qquad s, t = 1, \ldots, N_\mathcal{P}, \tag{4.45}$$

where $K_m \in \mathbb{R}^{N_X \times N_X}$ are finite element matrices with entries

$$[K_m]_{ij} := \int_D a_m(\mathbf{x}) \nabla \varphi_j(\mathbf{x}) \cdot \nabla \varphi_i(\mathbf{x}) d\mathbf{x}, \qquad i,j = 1,\dots,N_X,\ m = 0,\dots,M_{\mathcal{P}}, \tag{4.46}$$

whereas each block $\boldsymbol{F}_s \in \mathbb{R}^{N_X}$ of source vector $\boldsymbol{F} \in \mathbb{R}^{N_X N_{\mathcal{P}}}$ in (4.44) is given by

$$\boldsymbol{F}_s \ = \ (P_1, P_s)_\pi \boldsymbol{f}_0, \qquad s = 1,\dots,N_{\mathcal{P}},$$

where the vector $\boldsymbol{f}_0 \in \mathbb{R}^{N_X}$ is defined by

$$(\boldsymbol{f}_0)_i := \int_D f(\mathbf{x}) \varphi_i(\mathbf{x}) d\mathbf{x}, \qquad i = 1,\dots,N_X.$$

Notice that since $(P_1, P_s)_\pi = \delta_{1s}$, for all $s = 1,\dots,N_{\mathcal{P}}$, due to the orthogonality of polynomials $P_s \in \mathcal{P}_{\mathcal{P}}$, all blocks $\boldsymbol{F}_s$ in (4.44) are zero for all $s = 2,\dots,N_{\mathcal{P}}$; this is a simple consequence of the fact that we are considering non-parametric source terms in problem (4.5).

The linear system (4.44) can be further neatly expressed using Kronecker products of matrices and vectors. Let $G_0, G_m \in \mathbb{R}^{N_{\mathcal{P}} \times N_{\mathcal{P}}}$ be the *stochastic* matrices with entries

$$[G_0]_{st} := (P_s, P_t)_\pi = \delta_{st} \qquad \text{and} \qquad [G_m]_{st} := (y_m P_s, P_t)_\pi, \quad m = 1,\dots,M_{\mathcal{P}}, \tag{4.47}$$

for all $s,t = 1,\dots,N_{\mathcal{P}}$. Notice that G_0 is the $N_{\mathcal{P}} \times N_{\mathcal{P}}$ identity matrix. Then, we can rewrite the matrix $\boldsymbol{B}$ and vector $\boldsymbol{F}$ in (4.44) as

$$\boldsymbol{B} = G_0 \otimes K_0 + \sum_{m=1}^{M_{\mathcal{P}}} G_m \otimes K_m \qquad \text{and} \qquad \boldsymbol{F} = g_0 \otimes \boldsymbol{f}_0,$$

where g_0 is the first column of G_0 and $\otimes$ denotes the Kronecker product. We recall that, for a given matrix $A \in \mathbb{R}^{m \times n}$ and a matrix $B \in \mathbb{R}^{p \times q}$, with $m,n,p,q \in \mathbb{N}$, the Kronecker product $A \otimes B$ is the matrix of dimension $mp \times nq$ defined as

$$A \otimes B := \begin{pmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \dots & A_{mn}B \end{pmatrix}.$$

Unlike G_0, the matrices G_m defined by (4.47) are not diagonal but they are still highly sparse due to the orthogonality of polynomials of $\mathcal{P}_{\mathcal{P}}$. In particular, they can be computed from the three term recurrence (4.25) satisfied by the univariate orthonormal polynomials $p_k^m \in L^2_{\pi_m}(\Gamma_m)$ for each $m = 1,\dots,M_{\mathcal{P}}$.

Theorem 4.1. *Let Assumption 4.1 hold. Then, each matrix G_m defined in* (4.47) *has at most two*

nonzero entries per row:

$$[G_m]_{st} = \begin{cases} \beta^m_{\nu_m^{(s)}+1} & \text{if } \nu_m^{(s)} = \nu_m^{(t)} - 1 \text{ and } \nu_i^{(s)} = \nu_i^{(t)} \text{ for each } i \in \{1,\dots,M_{\mathcal{P}}\} \setminus m, \\ \beta^m_{\nu_m^{(s)}} & \text{if } \nu_m^{(s)} = \nu_m^{(t)} + 1 \text{ and } \nu_i^{(s)} = \nu_i^{(t)} \text{ for each } i \in \{1,\dots,M_{\mathcal{P}}\} \setminus m, \\ 0 & \text{otherwise,} \end{cases}$$

for $s,t = 1,\dots,N_{\mathcal{P}}$ *and where* $\beta^m_{\nu_m^{(s)}}$ *denote the coefficients from the three-term recurrence* (4.25) *associated with polynomials* $P_s \in \mathcal{P}_{\mathcal{P}}$.

For a proof of Theorem 4.1, see, e.g., [88, Theorem 9.59]. From definition (4.47), the block matrix (4.45) is given by

$$B_{ts} = [G_0]_{st} K_0 + \sum_{m=1}^{M_{\mathcal{P}}} [G_m]_{st} K_m, \qquad s,t = 1,\dots,N_{\mathcal{P}}.$$

Furthermore, Theorem 4.1 gives $[G_m]_{ss} = 0$, for all $s = 1,\dots,N_{\mathcal{P}}$. Then

$$B_{tt} = K_0 \qquad \text{and} \qquad B_{ts} = \sum_{m=1}^{M_{\mathcal{P}}} [G_m]_{ts} K_m, \ \text{ for } t \neq s.$$

Since both finite element matrices K_m in (4.46) and stochastic matrices G_m in (4.47) are sparse, the left-hand side matrix $\boldsymbol{B}$ in (4.44) turns out to be highly block sparse; see, e.g., [106], [107, Figure 3], and [88, Figure 9.10] for examples showing the block sparsity patterns of matrix $\boldsymbol{B}$ in case of complete polynomials (see Example 4.2.5). See also [65] for further details about the structure of linear systems arising from PDE problems with random data.

Once the Galerkin approximation (4.43) is obtained (by solving the associated linear system (4.44)), we can compute its mean and variance as follows.

Proposition 4.2. *Let* $u_{X\mathcal{P}}$ *in* (4.43) *be the stochastic Galerkin approximation in* $V_{X\mathcal{P}}$ *satisfying weak formulation* (4.36). *The mean value and the variance of* $u_{X\mathcal{P}}$ *are given by*

$$\mathbb{E}[u_{X\mathcal{P}}](x) = \phi_1(x) \qquad \text{and} \qquad \mathrm{Var}(u_{X\mathcal{P}})(x) = \sum_{s=2}^{N_{\mathcal{P}}} \phi_s(x)^2, \qquad x \in D. \tag{4.48}$$

Proof. The mean value simply follows by the orthogonality of polynomials $P_s \in \mathcal{P}_{\mathcal{P}}$ with respect to $(\cdot,\cdot)_\pi$ and since $P_1(y) = 1$ for all $y \in \Gamma$:

$$\mathbb{E}[u_{X\mathcal{P}}](x) := \int_\Gamma u_{X\mathcal{P}}(x,y)\, d\pi(y) = \phi_1(x) + \sum_{s=2}^{N_{\mathcal{P}}} \phi_s(x) \underbrace{(P_1,P_s)_\pi}_{=0} = \phi_1(x).$$

Similarly, we have that

$$\begin{aligned}\mathbb{E}[u_{X\mathcal{P}}^2](\mathbf{x}) &= \int_\Gamma \left(\phi_1^2(\mathbf{x}) + \left(\sum_{s=2}^{N_\mathcal{P}} \phi_s(\mathbf{x})P_s(\mathbf{y})\right)^2 + 2\phi_1(\mathbf{x})\sum_{s=2}^{N_\mathcal{P}} \phi_s(\mathbf{x})P_s(\mathbf{y})\right) d\pi(\mathbf{y}) \\ &= \phi_1^2(\mathbf{x}) + \sum_{s=2}^{N_\mathcal{P}} \phi_s^2(\mathbf{x})\underbrace{(P_s,P_s)_\pi}_{=1} + 2\sum_{\substack{s\neq r\\ s,r\neq 1}} \phi_s(\mathbf{x})\phi_r(\mathbf{x})\underbrace{(P_s,P_r)_\pi}_{=0} + 2\phi_1(\mathbf{x})\sum_{s=2}^{N_\mathcal{P}} \phi_s(\mathbf{x})\underbrace{(P_1,P_s)_\pi}_{=0} \\ &= \phi_1^2(\mathbf{x}) + \sum_{s=2}^{N_\mathcal{P}} \phi_s^2(\mathbf{x}).\end{aligned}$$

Since the variance is given by $\mathrm{Var}(u_{X\mathcal{P}}) = \mathbb{E}[u_{X\mathcal{P}}^2] - \mathbb{E}[u_{X\mathcal{P}}]^2$ (cf. (3.7)), this concludes the proof. □

The SGFEM became increasingly popular over the last decades although its computational cost is high. Due to approximations of both physical and parametric space simultaneously, the resulting discretisation requires the solution of linear system (4.44) (of $N_{X\mathcal{P}}$ equations) which is usually many orders of magnitude larger than the subproblems that can be solved in parallel by sampling-based methods such as Monte Carlo and stochastic collocation methods. This is indeed the most significant challenge associated with the SGFEM approach, i.e., the so-called 'curse of dimensionality': the linear system (4.44) may be huge, and its size grows fast with the size of the stochastic discretisation. For example, in Figure 4.1 we plot the dimensions $N_{X\mathcal{P}}$ of Galerkin linear systems (4.44) arising from discretisations of square domain $D = (0,1)^2$ using a fine triangulation of mesh-size $h = 2^{-5.5}$ and the set of complete polynomials associated with the index set $\mathcal{P}(M,n)$, on the left, and the set of tensor product polynomials associated with the index set $\widetilde{\mathcal{P}}(M,n)$, on the right (see Example 4.2.5). Observe that linear system (4.44) becomes infeasible to be solved on Desktop PCs that will go out of memory quickly for small values of the truncation parameter M.

Nevertheless, in dealing with the huge linear system (4.44), the resulting left-hand side matrix $\boldsymbol{B}$ is never fully assembled, in particular, if the random field is given as a linear function of the parameters (cf. (4.8)) and discretisations in the parameter space are done by means of orthogonal polynomials. In this case, $\boldsymbol{B}$ is highly block sparse and the linear system can be solved with much less effort than that suggested by its size. Due to the block structure and the sparsity pattern, it suffices to store the $N_\mathcal{P}$ finite element matrices K_m in (4.46) as well as the entries of each stochastic matrix G_m defined in (4.47), so that a careful use of matrix-vector multiplications of single blocks permits to solve the linear system (4.44) efficiently; see, e.g., [106]. Moreover, since the block matrix $\boldsymbol{B}$ in (4.44) is symmetric and positive definite (due to the positivity of the underlying random field, see Section 4.1.3), linear system (4.44) is thus perfectly suitable to be solved by iterative

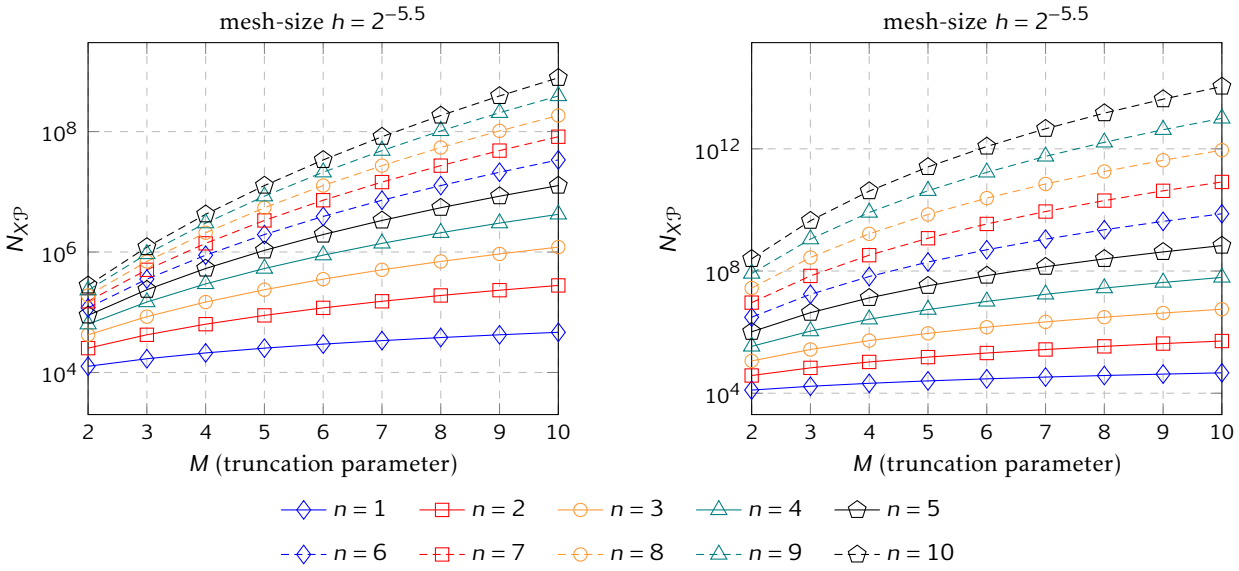


Figure 4.1. Dimension $N_{X\mathcal{P}}$ of Galerkin linear system (4.44) for discretisation on a conforming and shape-regular triangulation with mesh-size $h = 2^{-5.5}$ of the square domain $D = (0,1)^2$ and using complete polynomials associated with the index set $\mathcal{P}(M,n)$ (left) and tensor product polynomials associated with the index set $\widetilde{\mathcal{P}}(M,n)$ (right) defined in Example 4.2.5. Recall that the truncation parameter M also denotes the number of active parameters.

solvers such as the Conjugate Gradient Method. However, $\boldsymbol{B}$ is ill-conditioned with respect to both spatial and stochastic parameters. It is well known that single finite element matrices (4.46) are ill conditioned with respect to the mesh-size of the triangulation and then solving linear system (4.44) becomes more difficult as soon as the underlying triangulation is refined. Likewise, $\boldsymbol{B}$ is also ill-conditioned with respect to the number of (active) parameters as well as the (total) degree of polynomials in such parameters; see, e.g., [107, Lemma 3.7] and [88, pp. 410–412]. To overcome this problem, the construction of efficient preconditioners for linear system (4.44) turns out to be of fundamental importance.

For the implementation of efficient methods for solving linear systems arising from SGFEM approximations of PDEs with random inputs, we refer to [71, 106, 107] for mean-based preconditioned solvers and to [127] for solvers using Kronecker product preconditioners. In addition, see [128, 129] and [64, 122] for optimal preconditioned generalised minimum-residual (GMRES) and minimum-residual (MINRES) based solvers, respectively, and [108] for efficient reduced-basis solvers.

Chapter 5

Adaptive algorithms driven by hierarchical a posteriori error estimates

The theory of *a posteriori* error estimation for partial differential equations has become an essential ingredient especially for all those applications in which discretisations are made by finite element approximations. Since the pioneering use of a posteriori estimates by Babuška and Rheinboldt in [6, 7], the literature about this topic has hugely increased and many effective techniques are available today for the error estimation under the FEM context (see, e.g., [2, 131]). A posteriori error estimates not only provide valuable information about the accuracy of the approximation. Typically, such estimates are computed *locally*, hence they supply meaningful information about the distribution of the error among the elements forming the spatial discretisation. In this sense, a posteriori error estimates are effective usable indicators for local refinement schemes which are at the basis of adaptive FEM algorithms.

In this chapter, we consider the use of hierarchical bases for the a posteriori estimation of the energy norm of the error of the solution to problem (4.5). In the non-parametric setting, earlier use of *hierarchical* estimates dates back to [138, 137]. Later, this approach has been extensively investigated in, e.g., [16, 50, 15, 12]. In the parametric setting, the derivation of such estimates relies on the construction of enriched spaces via tensor products of Hilbert spaces. To our knowledge, the first application of hierarchical estimates in the context of PDEs with parametric uncertainty firstly appeared in [24]. Further developments in the same direction can be found in [29].

In what follows, we first recall the hierarchical error estimate introduced in [29]. Next, we describe the adaptive algorithm presented in recent work [27] for the energy error estimation of parametric problem (4.5). We do not analyse convergence properties of the proposed algorithm but we rather focus on its design and its novel error-reduction based version. The performance of

the algorithm is then showed by a set of numerical experiments where we consider representative examples of problem (4.5) having both spatially regular as well as singular solutions.

5.1 Enrichments via hierarchical basis

Let $u \in V$ be the solution to weak problem (4.17) and let $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ be the Galerkin approximation satisfying discrete weak formulation (4.36). Let $e := u - u_{X\mathcal{P}} \in V$ be the true error satisfying the following residual equation

$$B(e, v) \;=\; F(v) - B(u_{X\mathcal{P}}, v) \qquad \forall\, v \in V. \tag{5.1}$$

Notice that since $V_{X\mathcal{P}} \subset V$, the following *Galerkin orthogonality* holds:

$$B(e, v) \;=\; 0 \qquad \forall\, v \in V_{X\mathcal{P}}. \tag{5.2}$$

5.1.1 Enriched tensor product spaces

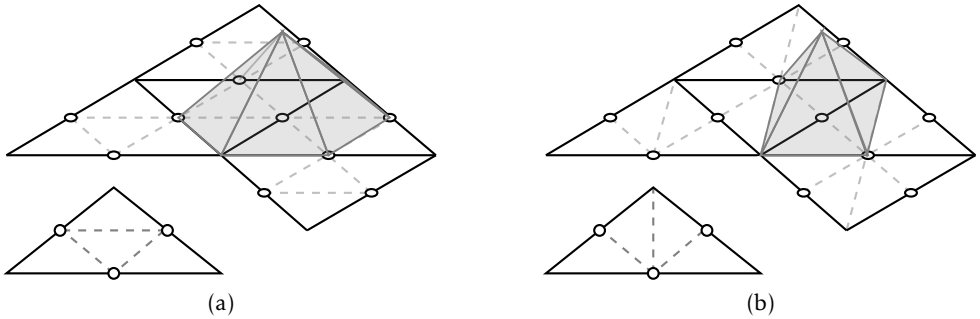
The approximation provided by $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ can be further improved by computing a Galerkin solution $\widehat{u}_{X\mathcal{P}}$ belonging to an enhanced finite-dimensional subspace $\widehat{V}_{X\mathcal{P}}$ of V such that $\widehat{V}_{X\mathcal{P}} \supset V_{X\mathcal{P}}$. This subspace can be constructed by enriching the finite element space $X \subset H_0^1(D)$ and/or the polynomial space $\mathcal{P}_{\mathcal{P}} \subset L^2_\pi(\Gamma)$.

Let $\widehat{X} \subset H_0^1(D)$ be an enriched finite element space such that $\widehat{X} \supset X$. The space $\widehat{X}$ is usually constructed by augmenting X with new functions which vanish at the element vertices of the triangulation $\mathcal{T}$ associated with X. For example, suppose that $X = \mathcal{S}_0^1(\mathcal{T})$ is a first-order finite element space on $\mathcal{T}$. We can add to X higher-order basis functions, e.g., piecewise quadratic basis on the same triangulation $\mathcal{T}$ (p-enrichment). Alternatively, X can be augmented with piecewise linear basis functions corresponding to vertices introduced by a uniform refinement of $\mathcal{T}$ (h-enrichment). In both cases, the enhanced space $\widehat{X}$ can be defined as

$$\widehat{X} := X \oplus Y \qquad \text{with} \qquad Y := \Big\{ v \in \widehat{X} \;:\; v(\mathbf{x}) = 0 \quad \forall\, \mathbf{x} \in \mathcal{N}(\mathcal{T}) \Big\}. \tag{5.3}$$

Here, the subspace $Y \subset H_0^1(D)$ is a finite element space such that $X \cap Y = \{0\}$ and it is called the *detail (finite element) space*; see Figure 5.1.

Remark 5.1.1. *Since $X \cap Y = \{0\}$ and $\langle A_0 \cdot, \cdot \rangle$ defines an inner product in $H_0^1(D)$ (here, A_0 is the symmetric operator defined in* (4.12)*), it is well known that there exists a positive constant $q_{\mathsf{cbs}} \in [0, 1)$*



(a) (b)

Figure 5.1. Enhancement via h-enrichment. (a) Piecewise linear basis function of Y associated with so called *red* (or *regular*) refinement of $\mathcal{T}$; (b) Piecewise linear basis function of Y associated with the uniform refinement made by three newest vertex bisections per element.

depending only on X and Y such that the strengthened Cauchy-Buniakowskii-Schwarz (CBS) inequality holds (see, e.g., [59] and [2, Theorem 5.4])

$$|\langle A_0 v, w\rangle| \leq q_{\mathsf{cbs}} \langle A_0 v, v\rangle^{1/2} \langle A_0 w, w\rangle^{1/2} \qquad \forall\, v \in X,\ \forall\, w \in Y. \tag{5.4}$$

Inequality (5.4) *is a key tool in some areas of numerical analysis as, in particular, it appears in the analysis of some types of a posteriori error estimators for finite element approximations of PDEs; see, e.g., [110, 46, 29].*

The polynomial space $\mathcal{P}_{\mathcal{P}} \subset L^2_\pi(\Gamma)$ can be enriched by augmenting the index set $\mathcal{P}$ with a new set of indices. The enriched polynomial space may consist of polynomials in new active parameters y_m and/or higher-order polynomials in the same active parameters in the Galerkin approximation. We introduce a finite index set $\mathcal{Q} \subset \mathcal{J}$ such that $\mathcal{P} \cap \mathcal{Q} = \emptyset$ and define

$$\widehat{\mathcal{P}} := \mathcal{P} \cup \mathcal{Q} \qquad \text{and} \qquad \widehat{\mathcal{P}_{\mathcal{P}}} := \mathcal{P}_{\mathcal{P}} \oplus \mathcal{P}_{\mathcal{Q}}, \tag{5.5}$$

where $\widehat{\mathcal{P}} \subset \mathcal{J}$ denotes the enriched index set and $\widehat{\mathcal{P}_{\mathcal{P}}}$ denotes the enriched polynomial space, with $\mathcal{P}_{\mathcal{Q}} := \operatorname{span}\{P_\mu : \mu \in \mathcal{Q}\}$ being the corresponding space generated by polynomials P_μ (see (4.29)), associated with indices $\mu \in \mathcal{Q}$, such that $\mathcal{P}_{\mathcal{P}} \cap \mathcal{P}_{\mathcal{Q}} = \{0\}$. The subset $\mathcal{Q}$ is called the *detail index set*. Since $\mathcal{Q} \subset \mathcal{J} \setminus \mathcal{P}$, we have

$$(P_\nu(\mathbf{y}), P_\mu(\mathbf{y}))_\pi = \prod_{m=1}^{\infty} \Big(p^m_{\nu_m}(y_m), p^m_{\mu_m}(y_m) \Big)_{\pi_m} = \prod_{m=1}^{\infty} \delta_{\nu_m \mu_m} = \delta_{\nu\mu} = 0, \tag{5.6}$$

for all $\nu \in \mathcal{P}$ and $\mu \in \mathcal{Q}$, i.e., $\mathcal{P}_{\mathcal{P}}$ and $\mathcal{P}_{\mathcal{Q}}$ are orthogonal with respect to inner product $(\cdot,\cdot)_\pi$.

With both finite element spaces $X, Y \subset H^1_0(D)$ and polynomial spaces $\mathcal{P}_{\mathcal{P}}, \mathcal{P}_{\mathcal{Q}} \subset L^2_\pi(\Gamma)$, we define the finite-dimensional tensor product spaces

$$V_{Y\mathcal{P}} := Y \otimes \mathcal{P}_{\mathcal{P}} \qquad \text{and} \qquad V_{X\mathcal{Q}} := X \otimes \mathcal{P}_{\mathcal{Q}}, \tag{5.7}$$

and the following enriched finite-dimensional space $\widehat{V}_{X\mathcal{P}} \subset V$,

$$\widehat{V}_{X\mathcal{P}} := V_{X\mathcal{P}} \oplus \Big(V_{Y\mathcal{P}} \oplus V_{X\mathcal{Q}} \Big). \tag{5.8}$$

Note that in (5.8), one of the spaces $V_{Y\mathcal{P}}$ and $V_{X\mathcal{Q}}$ may be empty. In particular, $V_{X\mathcal{P}}$ is enriched by adding a set of extra basis functions $\{\phi_\nu(\mathbf{x})P_\nu(\mathbf{y})\}$, where either $\phi_\nu \in Y$ and $P_\nu \in \mathcal{P}_\mathcal{P}$ (if only X is enriched) or $\phi_\nu \in X$ and $P_\nu \in \mathcal{P}_\mathcal{Q}$ (if only $\mathcal{P}_\mathcal{P}$ is enriched).

Due to the tensor product structure of $V_{Y\mathcal{P}}$ and $V_{X\mathcal{Q}}$ defined in (5.7), and since $\mathcal{P} \cap \mathcal{Q} = \emptyset$, there holds

$$B_0(v, w) = 0 \qquad \forall\, v \in V_{Y\mathcal{P}},\ \forall\, w \in V_{X\mathcal{Q}}, \tag{5.9}$$

i.e., $V_{Y\mathcal{P}}$ and $V_{X\mathcal{Q}}$ are orthogonal with respect to inner product $B_0(\cdot,\cdot)$. To verify this, it suffices to insert the polynomial chaos expansions of $v \in V_{Y\mathcal{P}}$ and $w \in V_{X\mathcal{Q}}$ (cf. (4.38)) in (5.9) and then use (5.6). On the other hand, the spaces $V_{X\mathcal{P}}$ and $V_{Y\mathcal{P}}$ are such that the following inequality holds (see [24, Lemma 3.1])

$$|B_0(u, v)| \le q_{\text{cbs}} \|u\|_{B_0} \|v\|_{B_0} \qquad \forall\, u \in V_{X\mathcal{P}},\ \forall\, v \in V_{Y\mathcal{P}},$$

where q_{cbs} is the constant appearing in (5.4).

5.1.2 Enhanced Galerkin solutions

Let $\widehat{u}_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$ be the Galerkin approximation satisfying the discrete weak formualation posed on $\widehat{V}_{X\mathcal{P}}$, i.e.,

$$B(\widehat{u}_{X\mathcal{P}}, v) = F(v) \qquad \forall\, v \in \widehat{V}_{X\mathcal{P}}. \tag{5.10}$$

Since $V_{X\mathcal{P}} \subseteq \widehat{V}_{X\mathcal{P}} \subset V$, the enhanced solution $\widehat{u}_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$ provides an approximation not worse than that provided by $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$, i.e.,

$$\|u - \widehat{u}_{X\mathcal{P}}\|_B = \inf_{v \in \widehat{V}_{X\mathcal{P}}} \|u - v\|_B \le \|u - u_{X\mathcal{P}}\|_B, \tag{5.11}$$

where the equality is the associated best approximation property for solution $\widehat{u}_{X\mathcal{P}}$ (cf. (4.37)). Also, there holds the following the Galerkin orthogonality

$$B(u - \widehat{u}_{X\mathcal{P}}, v) = 0 \qquad \forall\, v \in \widehat{V}_{X\mathcal{P}}, \tag{5.12}$$

with $u - \widehat{u}_{X\mathcal{P}} \in V$ representing the error due to the approximation $\widehat{u}_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$ (cf. (5.2)). Furthermore, the symmetry of the bilinear form $B(\cdot,\cdot)$ and Galerkin orthogonality (5.12) imply that

$$\|u - u_{X\mathcal{P}}\|_B^2 \;=\; \|u - \widehat{u}_{X\mathcal{P}}\|_B^2 \;+\; \|\widehat{u}_{X\mathcal{P}} - u_{X\mathcal{P}}\|_B^2 . \tag{5.13}$$

5.1.3 Saturation assumption

Under the current setting, we assume that a property stronger than (5.11) holds. As commonly done in the analysis of hierarchical a posteriori error estimation for non-parametric problems, it is assumed that the enhanced solution $\widehat{u}_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$ does indeed represent an approximation to u better than $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ in the following sense

$$\|u - \widehat{u}_{X\mathcal{P}}\|_B \;\leq\; q_{\mathrm{sat}} \|u - u_{X\mathcal{P}}\|_B \quad \text{with} \quad q_{\mathrm{sat}} \in [0,1). \tag{5.14}$$

Inequality (5.14) is called *saturation assumption* and it is equivalent to the standard saturation assumption assumed in non-parametric finite element analysis; see, e.g., [16, 15, 12, 2, 13].

Both (5.11) and (5.13) imply that (5.14) always holds for some saturation constant $q_{\mathrm{sat}} \leq 1$. The real contribution of the saturation assumption is then that q_{sat} is strictly less than one. Although in the literature it has the status of unproven hypothesis, in the deterministic case, this assumption may be quite realistic in many practical settings (see, e.g., [2, Section 5.2]). According to the given problem, the saturation assumption is normally observed for solutions computed on fine triangulations and may eventually fail to hold only for pairs of solutions associated with triangulations in 'preasymptotic ranges' (see [39]). In [53], for example, the deterministic saturation inequality has been proved to hold for two-dimensional Poisson problems using first-order FEM whenever *data oscillations* are small. In [93], the proof of convergence of h-adaptive FEM algorithms implicitly generalises the validation of saturation property to second-order linear elliptic PDEs. On the other hand, it is still an open problem in the parametric setting, to find the right assumptions on the data in (4.17) and identifying the enriching subspaces $Y \subset H_0^1(D)$ and $\mathcal{P}_{\mathcal{Q}} \subset L_\pi^2(\Gamma)$ for which (5.14) can be guaranteed (see [24, Remark 3.1]).

5.2 Hierarchical error estimate

In order to estimate the energy norm of the error $e = u - u_{X\mathcal{P}} \in V$ satisfying (5.1), let us consider the solution $\widehat{u}_{X\mathcal{P}} - u_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$ to the residual problem

$$B(\widehat{u}_{X\mathcal{P}} - u_{X\mathcal{P}}, v) \;=\; F(v) - B(u_{X\mathcal{P}}, v) \qquad \forall\, v \in \widehat{V}_{X\mathcal{P}} . \tag{5.15}$$

The following estimates hold

$$\|\widehat{u}_{X\mathcal{P}} - u_{X\mathcal{P}}\|_B^2 \leq \|e\|_B^2 \leq \frac{1}{1-q_{\text{sat}}^2}\|\widehat{u}_{X\mathcal{P}} - u_{X\mathcal{P}}\|_B^2. \tag{5.16}$$

Here, the lower bound follows by (5.13) while the upper bound follows by saturation assumption (5.14).

5.2.1 Error estimation using enriching subspaces

Error estimates (5.16) bear the computational cost associated with computing the enhanced Galerkin solution $\widehat{u}_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$. In addition, the evaluation of the norm $\|\cdot\|_B$ is expensive since it incorporates all coefficients a_m associated with all active parameters in $\text{supp}(\mathcal{P})$. To overcome these two problems, we recall the standard hierarchical approach firstly introduced in [24] and further developed in [29].

Let $\widehat{e}_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$ be the unique solution to the following residual problem

$$B_0(\widehat{e}_{X\mathcal{P}}, v) = F(v) - B(u_{X\mathcal{P}}, v) \qquad \forall\, v \in \widehat{V}_{X\mathcal{P}}. \tag{5.17}$$

Note the use of bilinear form $B_0(\cdot,\cdot)$ on the left-hand side of (5.17). The relation between $\widehat{e}_{X\mathcal{P}}$ and $\widehat{u}_{X\mathcal{P}} - u_{X\mathcal{P}}$ is given by

$$\lambda\|\widehat{e}_{X\mathcal{P}}\|_{B_0}^2 \leq \|\widehat{u}_{X\mathcal{P}} - u_{X\mathcal{P}}\|_B^2 \leq \Lambda\|\widehat{e}_{X\mathcal{P}}\|_{B_0}^2, \tag{5.18}$$

where λ and Λ are defined in (4.22) (see [24, Proposition 4.2]). Putting together the energy error estimates (5.16) and (5.18), we obtain

$$\lambda\|\widehat{e}_{X\mathcal{P}}\|_{B_0}^2 \leq \|e\|_B^2 \leq \frac{\Lambda^2}{1-q_{\text{sat}}^2}\|\widehat{e}_{X\mathcal{P}}\|_{B_0}^2. \tag{5.19}$$

Now, exploiting the tensor product structure of the enriched space $\widehat{V}_{X\mathcal{P}}$, consider the following two independent problems posed on the lower-dimensional subspaces $V_{Y\mathcal{P}}$ and $V_{X\mathcal{Q}}$ defined in (5.7),

$$\text{find} \quad e_{Y\mathcal{P}} \in V_{Y\mathcal{P}} \quad \text{s.t.} \quad B_0(e_{Y\mathcal{P}}, v) = F(v) - B(u_{X\mathcal{P}}, v) \qquad \forall\, v \in V_{Y\mathcal{P}}, \tag{5.20}$$

$$\text{find} \quad e_{X\mathcal{Q}} \in V_{X\mathcal{Q}} \quad \text{s.t.} \quad B_0(e_{X\mathcal{Q}}, v) = F(v) - B(u_{X\mathcal{P}}, v) \qquad \forall\, v \in V_{X\mathcal{Q}}. \tag{5.21}$$

Combining the *spatial* estimator $e_{Y\mathcal{P}} \in V_{Y\mathcal{P}}$ and the *parametric* estimator $e_{X\mathcal{Q}} \in V_{X\mathcal{Q}}$ satisfying (5.20) and (5.21), respectively, we introduce the following a posteriori error estimate (see [29]),

$$\eta_{X\mathcal{P}}^2 := \|e_{Y\mathcal{P}}\|_{B_0}^2 + \|e_{X\mathcal{Q}}\|_{B_0}^2. \tag{5.22}$$

Notice that $\eta_{X\mathcal{P}} = \| e_{Y\mathcal{P}} + e_{X\mathcal{Q}} \|_{B_0}$ due to the orthogonality of polynomial spaces $\mathcal{P}_\mathcal{P}$ and $\mathcal{P}_\mathcal{Q}$ with respect to inner product $(\cdot,\cdot)_\pi$. The estimate $\eta_{X\mathcal{P}}$ satisfies

$$\eta_{X\mathcal{P}}^2 \le \| \widehat{e}_{X\mathcal{P}} \|_{B_0}^2 \le \frac{1}{1-q_{\mathsf{cbs}}^2} \eta_{X\mathcal{P}}^2, \tag{5.23}$$

where q_{cbs} is the constant satisfying (5.4) (see [29, Lemma 4.1]). Therefore, putting together (5.19) and (5.23), $\eta_{X\mathcal{P}}$ is proved to be an efficient and reliable a posteriori estimate for the energy norm of the error.

Proposition 5.1 ([29, Theorem 4.1])**.** *Under saturation assumption* (5.14)*, the error estimate* $\eta_{X\mathcal{P}}$ *defined in* (5.22) *satisfies*

$$\lambda \eta_{X\mathcal{P}}^2 \le \| e \|_B^2 \le \frac{\Lambda}{(1-q_{\mathsf{sat}}^2)(1-q_{\mathsf{cbs}}^2)} \eta_{X\mathcal{P}}^2, \tag{5.24}$$

where λ *and* Λ *are the constants in* (4.21)*,* q_{cbs} *is the constant in* (5.4)*, and* q_{sat} *is the saturation constant in* (5.14)*.*

In the remainder of the thesis, we will refer to the a posteriori error estimate $\eta_{X\mathcal{P}}$ defined in (5.22) as *hierarchical* estimate.

Remark 5.2.1. *The parametric estimator* $e_{X\mathcal{Q}}$ *satisfying problem* (5.21)*, as well as its norm* $\| e_{X\mathcal{Q}} \|_{B_0}$*, can be computed from single contributions associated with individual indices in* $\mathcal{Q}$ *as follows:*

$$e_{X\mathcal{Q}} = \sum_{\mu \in \mathcal{Q}} e_{X\mathcal{Q}}^{(\mu)}, \qquad \| e_{X\mathcal{Q}} \|_{B_0}^2 = \sum_{\mu \in \mathcal{Q}} \| e_{X\mathcal{Q}}^{(\mu)} \|_{B_0}^2 . \tag{5.25}$$

Here, $e_{X\mathcal{Q}}^{(\mu)}$ *represents the parametric estimator in* $X \otimes \mathcal{P}_\mu$*, with* $\mathcal{P}_\mu := \mathsf{span}\{P_\mu(\mathsf{y})\}$ *for* $\mu \in \mathcal{Q}$*, satisfying*

$$B_0(e_{X\mathcal{Q}}^{(\mu)}, v) = F(v) - B(u_{X\mathcal{P}}, v) \qquad \forall\ v \in X \otimes \mathcal{P}_\mu, \tag{5.26}$$

(see [29, Lemma 4.2]). In particular, due to the orthogonality of spaces $\mathcal{P}_\mathcal{P}$ *and* $\mathcal{P}_\mathcal{Q}$ *(see* (5.6)*), we have* $F(v) = 0$ *for all* $v \in X \otimes \mathcal{P}_\mu$*. In fact,* $F(v)$ *incorporates the term* $(P_0, P_\mu)_\pi = \delta_{0\mu}$ *which is then equal to zero as* $0 \in \mathcal{P}$ *does not belong to* $\mathcal{Q}$*. Thus,* (5.26) *reduces to*

$$B_0(e_{X\mathcal{Q}}^{(\mu)}, v) = -B(u_{X\mathcal{P}}, v) \qquad \forall\ v \in X \otimes \mathcal{P}_\mu . \tag{5.27}$$

Remark 5.2.2. *In [24], the authors considered the larger enriched space,*

$$\widetilde{V}_{X\mathcal{P}} := \widehat{V}_{X\mathcal{P}} \oplus V_{Y\mathcal{Q}} = V_{X\mathcal{P}} \oplus \Big(V_{Y\mathcal{P}} \oplus V_{X\mathcal{Q}} \oplus V_{Y\mathcal{Q}} \Big).$$

where $V_{Y\mathcal{Q}} := Y \otimes \mathcal{P}_\mathcal{Q}$*. However, they have also empirically observed that the contributing enrichment due to the space* $V_{Y\mathcal{Q}}$ *does not lead to any qualitative improvement of saturation assumption* (5.14) *as*

well as the associated hierarchical a posteriori error estimate.

Remark 5.2.3 (Multi-level SGFEM)**.** *The error estimation technique described above can be naturally extended to more sophisticated so-called multi-level SGFEM approximations (see, e.g., [54, 47]). Multi-level SGFEMs work with discretisation spaces $V_{X\mathcal{P}}$ in which the spatial component has a 'multi-level' structure. That is, for each $\nu \in \mathcal{P}$, suppose that a sequence of finite element spaces $(X_\nu)_{\nu\in\mathcal{P}} \subset H_0^1(D)$ associated with potentially different triangulations $(\mathcal{T}_\nu)_{\nu\in\mathcal{P}}$ is available. Then, the discretisation space is defined as*

$$V_{X\mathcal{P}} := \bigoplus_{\nu\in\mathcal{P}} V_{X\mathcal{P}}^{(\nu)} \qquad \text{with} \qquad V_{X\mathcal{P}}^{(\nu)} := X_\nu \otimes \mathcal{P}_\nu \quad \forall\, \nu \in \mathcal{P}.$$

Here, $X_\nu := \operatorname{span}\{\varphi_j^{(\nu)} : j = 1,\dots,N_X^{(\nu)}\}$, where $N_X^{(\nu)} \in \mathbb{N}$ for all $\nu \in \mathcal{P}$, and then the PC expansion of the associated SGFEM solution becomes

$$u_{X\mathcal{P}}(\mathbf{x},\mathbf{y}) = \sum_{\nu\in\mathcal{P}} \phi_\nu(\mathbf{x})P_\nu(\mathbf{y}) \quad \text{with} \quad \phi_\nu(\mathbf{x}) = \sum_{j=1}^{N_X^{(\nu)}} u_j^{(\nu)}\varphi_j^{(\nu)}(\mathbf{x}) \in X_\nu, \quad u_j^{(\nu)} \in \mathbb{R},\ \mathbf{x}\in D,\ \mathbf{y}\in\Gamma.$$

The associated enriched finite-dimensional space can be defined as

$$\widehat{V}_{X\mathcal{P}} := \underbrace{\Big(\bigoplus_{\nu\in\mathcal{P}} \big(\widehat{X}_\nu \otimes \mathcal{P}_\nu\big)\Big)}_{=:V_{X\mathcal{P}}\oplus V_{Y\mathcal{P}}} \oplus \Big(\underbrace{X_{\widetilde{\nu}} \otimes \mathcal{P}_{\mathcal{Q}}}_{=:V_{X\mathcal{Q}}}\Big),$$

where $\widehat{X}_\nu$ are the enriched finite element spaces $\widehat{X}_\nu := X_\nu \oplus Y_\nu$ for appropriate detail spaces $Y_\nu \subset H_0^1(D)$, for all $\nu \in \mathcal{P}$, and $\widetilde{\nu}$ is one of the indices of $\mathcal{P}$. The main advantage of using multi-level SGFEMs is that, if implemented appropriately, they can be immune to the 'curse of dimensionality' since they generate sequences of approximations for which the associated energy errors decay with the same rate as in case of corresponding non-parametric problems (see, e.g., [44, 45, 77]). For instance, for first-order finite elements, this rate is $\mathcal{O}(N^{-1/2})$, where N is the number of total degrees of freedom of the discretisation (cf. the numerical experiments in Section 5.4); see, e.g., [47] for details on stochastic Galerkin linear systems arising from multi-level SGFEMs as well as for the design of adaptive algorithms that are driven by hierarchical a posteriori error estimates.

5.2.2 Estimates of the error reduction

Proposition 5.1 shows that $\eta_{X\mathcal{P}}$ defined in (5.22) can be used to control the error in the Galerkin approximation. However, it also turned out that the component estimators $e_{Y\mathcal{P}} \in V_{Y\mathcal{P}}$ and $e_{X\mathcal{Q}} \in V_{X\mathcal{Q}}$ contributing to $\eta_{X\mathcal{P}}$ play an important role in an adaptive process that aims at reducing

the error until some tolerance is met. In particular, it has been shown in [29] that $\|e_{Y\mathcal{P}}\|_{B_0}$ (resp. $\|e_{X\mathcal{Q}}\|_{B_0}$) 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 $\mathcal{P}_\mathcal{P}$) and to compute the corresponding enhanced approximation. For example, suppose that the space $\widehat{V}_{X\mathcal{P}}$ in (5.8) is constructed by only enriching the polynomial space $\mathcal{P}_\mathcal{P}$, i.e., let $u_{X\widehat{\mathcal{P}}} \in V_{X\widehat{\mathcal{P}}}$ be enhanced approximation satisfying

$$B(u_{X\widehat{\mathcal{P}}}, v) = F(v) \qquad \forall\, v \in V_{X\widehat{\mathcal{P}}} := V_{X\mathcal{P}} \oplus V_{X\mathcal{Q}}. \tag{5.28}$$

Then, similarly to (5.13), there holds

$$\|u - u_{X\mathcal{P}}\|_B^2 = \|u - u_{X\widehat{\mathcal{P}}}\|_B^2 + \|u_{X\widehat{\mathcal{P}}} - u_{X\mathcal{P}}\|_B^2.$$

This shows that the error reduction achieved by enriching only $\mathcal{P}_\mathcal{P}$ is given by $\|u_{X\widehat{\mathcal{P}}} - u_{X\mathcal{P}}\|_B$. The same argument applies if we were to enrich only the finite element space X by computing the corresponding approximation $u_{\widehat{X}\mathcal{P}} \in V_{\widehat{X}\mathcal{P}}$ satisfying

$$B(u_{\widehat{X}\mathcal{P}}, v) = F(v) \qquad \forall\, v \in V_{\widehat{X}\mathcal{P}} := V_{X\mathcal{P}} \oplus V_{Y\mathcal{P}}. \tag{5.29}$$

In this case, the error reduction is given by the quantity $\|u_{\widehat{X}\mathcal{P}} - u_{X\mathcal{P}}\|_B$. The following two-sides bounds for both error reductions are proved in [24, Theorem 5.1]:

$$\lambda \|e_{Y\mathcal{P}}\|_{B_0}^2 \le \|u_{\widehat{X}\mathcal{P}} - u_{X\mathcal{P}}\|_B^2 \le \frac{\Lambda}{1 - q_{\mathrm{cbs}}^2} \|e_{Y\mathcal{P}}\|_{B_0}^2, \tag{5.30}$$

$$\lambda \|e_{X\mathcal{Q}}\|_{B_0}^2 \le \|u_{X\widehat{\mathcal{P}}} - u_{X\mathcal{P}}\|_B^2 \le \Lambda \|e_{X\mathcal{Q}}\|_{B_0}^2. \tag{5.31}$$

It is worth noticing that given the problem data and the computed Galerkin approximation $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$, the estimates $\|e_{Y\mathcal{P}}\|_{B_0}$ and $\|e_{X\mathcal{Q}}\|_{B_0}$ are computable for any finite-dimensional space $Y \subset H_0^1(D)$ and finite index set $\mathcal{Q} \subset \mathcal{I}$. Therefore, this means that both error reductions $\|u_{\widehat{X}\mathcal{P}} - u_{X\mathcal{P}}\|_B$ and $\|u_{X\widehat{\mathcal{P}}} - u_{X\mathcal{P}}\|_B$ can be estimated before the corresponding enhanced approximations are effectively computed (see (5.30) and (5.31)). In particular, we stress that the choice of the detail subspace Y and the detail index set $\mathcal{Q}$ may depend on whether we want to estimate the error in the Galerkin approximation $u_{X\mathcal{P}}$ or estimate the error reduction achieved by enhancing this approximation. For example, in order to obtain an accurate estimate of $\|e\|_B$, we should use 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 $\mathcal{Q} \subset \mathcal{I} \setminus \mathcal{P}$. How to choose a suitable index set $\mathcal{Q}$ is discussed in the next section.

Remark 5.2.4. *Due to decomposition* (5.25) *of parametric estimator* $e_{X\mathcal{Q}}$ *and error bounds* (5.31)*, the quantity* $\|e^{(\mu)}_{X\mathcal{Q}}\|_{B_0}$ *for* $\mu \in \mathcal{Q}$*, provides an estimate of the energy error reduction achieved by computing the Galerkin approximation* $u_{X\widetilde{\mathcal{P}}_\mu}$ *belonging to the space* $V_{X\widetilde{\mathcal{P}}_\mu} := V_{X\mathcal{P}} \oplus (X \otimes \mathcal{P}_\mu)$*, i.e., if the polynomial space* $\mathcal{P}_\mathcal{P}$ *is enriched by adding only the polynomial* $P_\mu \in \mathcal{P}_\mu$*. Hence, there holds*

$$\lambda \|e^{(\mu)}_{X\mathcal{Q}}\|^2_{B_0} \le \|u_{X\widetilde{\mathcal{P}}_\mu} - u_{X\mathcal{P}}\|^2_B \le \Lambda \|e^{(\mu)}_{X\mathcal{Q}}\|^2_{B_0} \qquad \forall\, \mu \in \mathcal{Q}. \tag{5.32}$$

5.2.3 Suitable detail index sets

An important aspect of the design of an efficient adaptive algorithm driven by hierarchical estimate $\eta_{X\mathcal{P}}$ is the need to account for those indices $\mu \in \mathcal{Q}$, if any, for which the corresponding estimator $e^{(\mu)}_{X\mathcal{Q}}$ is zero. In such case, also the associated error reduction is zero (see Remark 5.2.4). Then, computational efforts should be only directed to solve individual problems (5.27) yielding nonzero contributions to the error estimator $e_{X\mathcal{Q}} \in V_{X\mathcal{Q}}$ (recall that $e_{X\mathcal{Q}}$ can be decomposed into single contributing estimators $e^{(\mu)}_{X\mathcal{Q}}$ for all $\mu \in \mathcal{Q}$; see Remark 5.2.1).

Let us introduce some notation first. For $m \in \mathbb{N}$, let $\varepsilon^{(m)} := (\varepsilon^{(m)}_1, \varepsilon^{(m)}_2, \dots) \in \mathcal{J}$ be the Kronecker delta sequence such that $\varepsilon^{(m)}_k = \delta_{mk}$ for all $k \in \mathbb{N}$. Hereafter, we make the following assumption on the finite index set $\mathcal{P}$ used for parametric SGFEM discretisations. We assume that $\mathcal{P} \subset \mathcal{J}$ is a *monotone*[1] finite index set, i.e., for all $\nu \in \mathcal{P}$, the indices $\mu = \nu - \varepsilon^{(m)}$ belong to $\mathcal{P}$ for all $m \in \operatorname{supp}(\mathcal{P})$. Now, consider the following (infinite) index set

$$\partial\mathcal{P} := \left\{ \mu \in \mathcal{J} \setminus \mathcal{P} \;:\; \mu = \nu + \varepsilon^{(m)} \quad \forall\, \nu \in \mathcal{P},\, \forall\, m \in \mathbb{N} \right\}, \tag{5.33}$$

called the *boundary* of $\mathcal{P}$. Indices $\mu \in \partial\mathcal{P}$ are called the *neighbours* of indices in $\mathcal{P}$. Notice that $\operatorname{supp}(\mathcal{P} \cup \partial\mathcal{P}) = \mathbb{N}$ and that $\mathcal{P} \cup \partial\mathcal{P}$ is monotone. It can be shown that for any index set $\mathcal{Q} \subset \mathcal{J} \setminus (\mathcal{P} \cup \partial\mathcal{P})$ the error estimator $e_{X\mathcal{Q}}$ is identically zero (see [29, Lemma 4.3]). That is, for all indices μ belonging to such $\mathcal{Q}$, the estimator $e^{(\mu)}_{X\mathcal{Q}}$ is zero and no error reduction is expected from adding μ into the parametric discretisation (cf. (5.32)). Moreover, [29, Corollary 4.1] shows that this also holds even in the case of parametric right-hand side sources $f(\mathbf{y})$ in problem (4.5) with affine dependence on parameters (cf. (4.8)).

Remark 5.2.5. *Let us emphasise that the monotonicity of the index set* $\mathcal{P}$ *is not strictly required from the discretisation point of view nor for the design of an adaptive algorithm. It is rather an algorithmically desirable property which allows neighbour indices to be accessed easily and that ensures a kind of 'tree-*

[1] Monotone sets are also often called *downward closed* sets or *lower* sets (see, e.g., [55]).

structure' for the polynomial degrees of active parameters in the index set.

Example 5.2.1. *Consider the set of complete polynomials $\mathcal{P}_{\mathcal{P}(2,1)}$ associated with the index set $\mathcal{P}(2,1)$, both defined in Example 4.2.5. Let us consider the following enriched polynomial space*

$$\mathcal{P}_{\mathcal{P}(4,2)} := \mathcal{P}_{\mathcal{P}(2,1)} \oplus \mathcal{P}_{\mathcal{Q}},$$

associated with the index set $\mathcal{P}(4,2)$, where the detail index set $\mathcal{Q} \subset \mathcal{J}$ is given by $\mathcal{Q} = \mathcal{P}(4,2) \setminus \mathcal{P}(2,1)$. That is,

$$\mathcal{P}(2,1) = \begin{Bmatrix} (0,0) \\ (1,0) \\ (0,1) \end{Bmatrix} \quad \text{and} \quad \mathcal{Q} = \begin{Bmatrix} (0,0,0,1), & (0,0,1,0), & (1,0,0,1) & (\mathbf{0},\mathbf{0},\mathbf{1},\mathbf{1}) \\ (1,0,1,0), & (0,1,0,1), & (1,1,0,0) & (\mathbf{0},\mathbf{0},\mathbf{2},\mathbf{0}) \\ (0,1,1,0), & (2,0,0,0), & (0,2,0,0) & (\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{2}) \end{Bmatrix}.$$

In particular, $\#\mathcal{Q} = \#\mathcal{P}(4,2) - \#\mathcal{P}(2,1) = 15 - 3 = 12$. However, from [29, Lemma 4.3], the number of indices $\mu \in \mathcal{Q}$ associated with nonzero error estimators $e_{X_{\mathcal{Q}}}^{(\mu)}$ is only 9. In fact, notice that the boldface indices of $\mathcal{Q}$ do not belong to the boundary $\partial\mathcal{P}(2,1)$ since they cannot be obtained by adding Kronecker delta sequences to the indices in $\mathcal{P}(2,1)$ (see also [29, Example 4.1]).

The above discussion suggests to consider only the finite detail index sets $\mathcal{Q}$ extracted from the boundary $\partial\mathcal{P}$ defined in (5.33).

Remark 5.2.6. *Lemma 4.3 and Corollary 4.1 in [29] do hold for more general boundary index sets $\partial\mathcal{P}$ defined as (see [29, Eq. (4.26)])*

$$\widetilde{\partial\mathcal{P}} := \left\{ \mu \in \mathcal{J} \setminus \mathcal{P} \ : \ \mu = \nu + \varepsilon^{(m)} \ \ \text{or} \ \ \mu = \nu - \varepsilon^{(m)} \quad \forall\, \nu \in \mathcal{P},\ \forall\, m \in \mathbb{N} \right\}.$$

Note that if $\mathcal{P}$ is monotone, then $\widetilde{\partial\mathcal{P}} = \partial\mathcal{P}$ since indices μ expressed as $\nu - \varepsilon^{(m)}$ are already in $\mathcal{P}$ due to its monotonicity.

5.3 Adaptive SGFEM algorithm

In this section, we describe the adaptive SGFEM algorithm presented in [27] for the energy error estimation of the parametric model problem (4.5). The algorithm is driven by hierarchical a posteriori error estimates (5.22) and it follows the standard finite element loop (2.13). Furthermore, it can be extended, in an appropriate way, to other parametric PDE problems with affine dependence on random parameters.

A novelty in this adaptive algorithm is how the balance between spatial and stochastic appro-

ximations is ensured. We describe two versions of the algorithm based on two different marking criteria adopted in the MARK module of the adaptive loop. It is common to perform either spatial or stochastic refinements of the discretisation space at each iteration of the algorithm, that is, either a local mesh-refinement of the underlying triangulation or a parametric enrichment of the index set is only pursued for the computation of enhanced approximations. Traditionally, the choice between these two refinements is based on the dominant error estimator contributing to the *total* error estimate (cf. [54, 55, 29, 56]); this is one possible version of the algorithm. An alternative strategy is implemented in the second version: here, the refinement type is chosen by comparing the error reduction estimates associated with *marked* elements and *marked* indices for spatial parametric approximations, respectively.

Before discussing these aspects of the algorithm, let us provide further details on the components of the adaptive loop. Below, we consider the case of spatial discretisation made by first-order finite element spaces, i.e., we let $X := \mathcal{S}^1_0(\mathcal{T})$.

5.3.1 Computation of spatial and parametric estimates

The computation of hierarchical estimate $\eta_{X\mathcal{P}}$ defined in (5.22) requires solving the problems for the estimators $e_{Y\mathcal{P}} \in V_{Y\mathcal{P}}$ and $e_{X\mathcal{Q}} \in V_{X\mathcal{Q}}$ satisfying (5.20) and (5.21), respectively. Suppose that the Galerkin solution $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ to problem (4.36) is available. Let us describe how the algorithm computes the estimators $e_{Y\mathcal{P}}$ and $e_{X\mathcal{Q}}$.

Computation of the spatial contribution

Let $e_{Y\mathcal{P}} \in V_{Y\mathcal{P}}$ be the spatial estimator satisfying (5.20). We choose the detail space $Y \subset H^1_0(D)$ to be associated with the uniform refinement of triangulation $\mathcal{T}$ obtained by three NVB refinements per element of $\mathcal{T}$. That is, $Y := \operatorname{span}\{\psi_j : j = 1,\dots,\#\mathcal{E}^\circ(\mathcal{T})\}$, where ψ_j denotes the piecewise linear Lagrange basis function associated with the midpoint of the j-th interior edge $E_j \in \mathcal{E}^\circ(\mathcal{T})$ such that $\psi_j(z_i) = \delta_{ij}$ with z_i being the midpoint of $E_i \in \mathcal{E}^\circ(\mathcal{T})$, $i,j = 1,\dots,\#\mathcal{E}^\circ(\mathcal{T})$ (see Figure 5.1(b)). Note that all ψ_j also vanish at vertices of $\mathcal{T}$ (cf. (5.3)).

Problem (5.20) is solved using a standard element residual technique (see [2, Section 3.3] and Remark 5.3.1 below). Specifically, on each element $T \in \mathcal{T}$, we compute a *local* spatial error

estimator by solving the local residual problem associated with (5.20): find $e_{Y\mathcal{P}}^{(T)} \in V_{Y\mathcal{P}}|_T$ such that

$$B_{0,T}(e_{Y\mathcal{P}}^{(T)}, v) = F_T(v) + \int_\Gamma \int_T \nabla \cdot (a(\mathbf{x},\mathbf{y}) \nabla u_{X\mathcal{P}}(\mathbf{x},\mathbf{y}))\, v(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\pi(\mathbf{y}) - \frac{1}{2} \sum_{E \in \mathcal{E}^\circ(T)} \int_\Gamma \int_E a(s,\mathbf{y}) [\![\nabla u_{X\mathcal{P}}]\!]_E\, v(s,\mathbf{y})\, ds\, d\pi(\mathbf{y}), \tag{5.34}$$

for all $v \in V_{Y\mathcal{P}}|_T$. Here, $V_{Y\mathcal{P}}|_T := Y|_T \otimes \mathcal{P}_\mathcal{P}$, where $Y|_T$ is the restriction of Y to the element $T \in \mathcal{T}$, $B_{0,T}$ and F_T denote the bilinear form and the linear functional restricted on Y, respectively, and $[\![\cdot]\!]_E$ denotes the flux jump across the edge E of T, i.e., for every $v \in X$,

$$[\![\nabla v]\!]_E := \nabla v|_T \cdot \boldsymbol{n} - \nabla v|_{T'} \cdot \boldsymbol{n} \qquad \forall\, E \in \mathcal{E}^\circ(T),$$

with $T' \in \mathcal{T}$ being the neighbour of T such that $E = T \cap T'$ and where $\boldsymbol{n}$ denotes the outward pointing unit normal vector to the edge $E \in \mathcal{E}^\circ(\mathcal{T})$. Note that local problems (5.34) are well-defined; in fact, $B_{0,T}(\cdot,\cdot)$ is coercive over T as $B_{0,T}(v,v) > 0$ for all $v \in V_{Y\mathcal{P}}|_T$.

One important feature of this error estimation technique is that the linear algebra associated with problem (5.34) is simple. In fact, due to the use of bilinear form $B_0(\cdot,\cdot)$ which does not incorporate the parameters (see (4.19)), the resulting left-hand side matrix of the linear system arising from (5.34) is the block diagonal matrix $\boldsymbol{B}_T \in \mathbb{R}^{3N_\mathcal{P} \times 3N_\mathcal{P}}$ given by the Kronecker product of the identity matrix G_0 (see (4.47)) and the reduced finite element matrix $K_T \in \mathbb{R}^{3\times 3}$ associated with $Y|_T$, i.e., for all $T \in \mathcal{T}$,

$$\boldsymbol{B}_T = G_0 \otimes K_T \qquad \text{with} \qquad [K_T]_{ij} = \int_T a_0(\mathbf{x}) \nabla \psi_j(\mathbf{x}) \cdot \nabla \psi_i(\mathbf{x}) d\mathbf{x}, \quad i,j = 1,\dots,3.$$

Here, $\{\psi_i\}_{i=1}^3$ denote the first-order Lagrange basis of $Y|_T$ (with $\dim(Y|_T) = 3$); notice that the indices i and j refer to the *local* enumeration of the basis of Y restricted to $T \in \mathcal{T}$. As a result, the numerical computation of local error estimators $e_{Y\mathcal{P}}^{(T)}$, for $T \in \mathcal{T}$, can be easily parallelised.

Remark 5.3.1 (Implicit estimators)**.** *The definition of problem* (5.34) *posed on single elements $T \in \mathcal{T}$ is a technique known as element residual method which dates back to [16]. This method is a specific error estimation approach belonging to the family of methods referred to as* implicit *error estimators; see, e.g., [2, Chapter 3] and [131, Section 1.7]. Contrary to explicit estimators (i.e., residual-based estimators, see, e.g., [2, Chapter 2] and [131, Section 1.4]) that are computable from data problem and the Galerkin approximation, implicit estimators require the solution of auxiliary local boundary value problems that approximate appropriately the single global residual equation (see* (5.17)*). The associated error estimate is then obtained by summing the norms of all local contributing estimators over the domain (see* (5.37)

below).

Remark 5.3.2. *In spite of global reliability and efficiency* (5.24) *of hierarchical error estimates, it is well known that the associated local error estimators (see* (5.34)*) are, in general, not 'reliable' in the sense that so-called effectivity indices (defined as the ratio of the total error estimate to the true error in the energy norm, see* (5.43)*) may become less than unity (see, e.g., [16] and [61, Section 1.5.2], as well as the results of numerical experiments in Section 5.4.1).*

Computation of the parametric contribution

Consider now the parametric estimator $e_{X\mathcal{Q}} \in V_{X\mathcal{Q}}$ satisfying (5.21). In addition to monotonicity, the finite detail index sets $\mathcal{Q}$ that we are going to define below, and that are used by the adaptive algorithm, are based on the assumption that the finite index sets $\mathcal{P}$ are ordered (see Section 4.2.2 and (4.42)).

As explained in Section 5.2.3, it is worth considering only those detail index sets for which $e_{X\mathcal{Q}}$ is nonzero. Such detail index sets are all subsets of $\partial\mathcal{P}$ defined in (5.33). Therefore, for a fixed $M_{\mathcal{Q}} \in \mathbb{N}$, we consider the finite detail index set $\mathcal{Q} \subset \partial\mathcal{P}$ for the computation of $e_{X\mathcal{Q}}$ as

$$\mathcal{Q} := \left\{ \mu \in \mathcal{I} \setminus \mathcal{P} \ : \ \mu = \nu + \varepsilon^{(m)} \quad \forall\, \nu \in \mathcal{P},\ m = 1, \dots, M_{\mathcal{P}} + M_{\mathcal{Q}} \right\}. \tag{5.35}$$

Here, the index set $\mathcal{Q}$ contains only those neighbours of the indices of $\mathcal{P}$ that have up to $M_{\mathcal{P}} + M_{\mathcal{Q}}$ active parameters, i.e., $M_{\mathcal{Q}}$ parameters more than those currently active in $\mathcal{P}$. In particular, the cardinality of $\mathcal{Q}$ in (5.35) is at most $N_{\mathcal{P}}(M_{\mathcal{P}} + M_{\mathcal{Q}})$ and the monotonocity of the enriched index set $\mathcal{P} \cup \mathcal{Q}$ is preserved. Notice that values of $M_{\mathcal{Q}}$ larger than 1 may lead to too large computational efforts when running the adaptive algorithms (see the numerical experiment in Section 7.3.2).

With the finite detail index set (5.35), we compute the individual parametric estimators $e^{(\mu)}_{X\mathcal{Q}} \in X \otimes \mathcal{P}_\mu$ by solving the linear systems arising from problems (5.27) for all $\mu \in \mathcal{Q}$. In particular, the corresponding left-hand side matrices $B^\mu \in \mathbb{R}^{N_X \times N_X}$ of such linear systems are given by the finite element matrices K_0 associated with the space X and corresponding to the parameter-free term $a_0(\mathsf{x})$ (see (4.46)). That is, B^μ is the same matrix for all $\mu \in \mathcal{Q}$, hence it has to be assembled only once.

Remark 5.3.3. *By considering ordered index sets $\mathcal{P}$, we are assuming that the initial parametric discretisations do not activate a parameter y_m without also activating all parameters y_n for all $1 \le n \le m$. The reason for this is the assumption that a parameter y_n is more 'important' than a parameter y_m*

($n \le m$), in the sense that y_m contributes less to expansion (4.8) *of the random field (recall the assumption on the magnitudes $\|\cdot\|_{L^\infty(D)}$ of spatial coefficients $(a_m)_{m\in\mathbb{N}}$ in Assumption 4.2). For example, consider the following monotone but not ordered index set:*

$$\mathcal{P} = \Big\{ \mathbf{0},\ (1,0,0),\ (0,0,1) \Big\}. \tag{5.36}$$

Here, $\mathrm{supp}(\mathcal{P}) = \{1,3\}$, i.e., there are $M_{\mathcal{P}} = \#\mathrm{supp}(\mathcal{P}) = 2$ active parameters which are not, however, y_1 and y_2 (but y_1 and y_3). In particular, coefficient a_3 would be active in expansion (4.8) *of the random field although $\|a_3\|_{L^\infty(D)} \le \|a_2\|_{L^\infty(D)}$.*

We stress that the order constraint is only, effectively, applied to the very first initial index set $\mathcal{P}$, since its monotonicity automatically implies that the enriched index sets are also ordered. If the initial index sets do not need to be ordered, we could modify (5.35) *by substituting $M_{\mathcal{P}}$ with a more general counter parameter $\widetilde{M}_{\mathcal{P}}$ defined as follows (see [29, Eq. (5.2)]):*

$$\widetilde{M}_{\mathcal{P}} := \begin{cases} 0 & \text{if } \mathcal{P} = \{\mathbf{0}\}, \\ \max\Big\{ \max(\mathrm{supp}(\nu)),\ \ \nu \in \mathcal{P}\setminus\{\mathbf{0}\} \Big\} & \text{otherwise.} \end{cases}$$

For index set (5.36)*, we have $\widetilde{M}_{\mathcal{P}} = 3 \neq 2 = M_{\mathcal{P}}$; however, notice that $\widetilde{M}_{\mathcal{P}} = M_{\mathcal{P}}$ for ordered index sets.*

Local error contributions

Once all local spatial estimators $e^{(T)}_{Y\mathcal{P}}$ satisfying (5.34) are computed for all $T \in \mathcal{T}$ and all parametric estimators $e^{(\mu)}_{X\mathcal{Q}}$ satisfying (5.27) are computed for all $\mu \in \mathcal{Q}$ (with $\mathcal{Q}$ given by (5.35)), we define the spatial estimate

$$\eta_{Y\mathcal{P}}(\mathcal{T})^2 := \sum_{T\in\mathcal{T}} \eta_{Y\mathcal{P}}(T)^2 \qquad \text{with} \qquad \eta_{Y\mathcal{P}}(T) := \|e^{(T)}_{Y\mathcal{P}}\|_{B_{0,T}} \qquad \forall\ T \in \mathcal{T}, \tag{5.37}$$

and the parametric estimate

$$\eta_{X\mathcal{Q}}(\mathcal{Q})^2 := \sum_{\mu\in\mathcal{Q}} \eta_{X\mathcal{Q}}(\mu)^2 \qquad \text{with} \qquad \eta_{X\mathcal{Q}}(\mu) := \|e^{(\mu)}_{X\mathcal{Q}}\|_{B_0} \qquad \forall\ \mu \in \mathcal{Q}. \tag{5.38}$$

Notice that while $\|e_{X\mathcal{Q}}\|_{B_0} = \eta_{X\mathcal{Q}}(\mathcal{Q})$ due to (5.25), we only have $\|e_{Y\mathcal{P}}\|_{B_0} \approx \eta_{Y\mathcal{P}}(\mathcal{T})$ due to the use of the implicit element residual problem technique (see [16, 2]). Therefore, we approximate the *total* hierarchical error estimate $\eta_{X\mathcal{P}}$ in (5.22) via

$$\eta_{X\mathcal{P}} \ \approx\ \eta := \Big(\eta_{Y\mathcal{P}}(\mathcal{T})^2 + \eta_{X\mathcal{Q}}(\mathcal{Q})^2 \Big)^{1/2}. \tag{5.39}$$

5.3.2 Marking strategy and refinements

In order to compute more accurate Galerkin solutions, enriched approximation spaces need to be constructed at each iteration of the adaptive loop. To this end, a marking strategy is needed to select a subset $\mathcal{M} \subseteq \mathcal{T}$ of elements to be refined or a subset $\mathcal{M} \subseteq \mathcal{Q}$ of indices to be included in the parametric approximation. In both cases, the algorithm employs the Dörfler marking strategy (see Strategy 2.2) to build the two subsets $\mathcal{M} \subseteq \mathcal{T}$ and $\mathcal{M} \subseteq \mathcal{Q}$ with minimal cardinality satisfying

$$\eta_{Y\mathcal{P}}(\mathcal{M})^2 := \sum_{T\in\mathcal{M}} \eta_{Y\mathcal{P}}(T)^2 \geq \theta_X \eta_{Y\mathcal{P}}(\mathcal{T})^2 \qquad \text{and} \qquad \eta_{X\mathcal{Q}}(\mathcal{M})^2 := \sum_{\mu\in\mathcal{M}} \eta_{X\mathcal{Q}}(\mu)^2 \geq \theta_{\mathcal{P}} \eta_{X\mathcal{Q}}(\mathcal{Q})^2, \quad (5.40)$$

respectively. Here, $\eta_{Y\mathcal{P}}(\mathcal{T})$ and $\eta_{X\mathcal{Q}}(\mathcal{Q})$ are the total estimates defined by (5.37) and (5.38) and $\theta_X, \theta_{\mathcal{P}} \in (0,1]$ are the corresponding marking parameters.

Once the sets $\mathcal{M} \subseteq \mathcal{T}$ and $\mathcal{M} \subseteq \mathcal{Q}$ are available, the algorithm can construct an enhanced discretisation space. While parametric enrichments are simply made by adding the set $\mathcal{M}$ to the index set $\mathcal{P}$, a refinement rule has to be set up for spatial mesh-refinements. To this end, the algorithm returns a new conforming triangulation by implementing the NVB refinements where reference edges are the longest edges of each element of $\mathcal{T}$ (see Section 2.3.2). Recall that the use of this mesh-refinement technique ensures that the finite element spaces associated with refined triangulations are nested (see Remark 2.3.1).

5.3.3 Adaptive loop

We now describe the loop of the proposed adaptive SGFEM algorithm. Throughout, we use the subscript (or superscript) $\ell \in \mathbb{N}_0$ for triangulations, index sets, Galerkin solutions, etc., associated with the ℓ-th iteration of the loop.

Starting with a conforming coarse triangulation $\mathcal{T}_0$ and an initial index set $\mathcal{P}_0$, at each iteration $\ell \in \mathbb{N}_0$, the finite element space $X_\ell := \mathcal{S}^1_0(\mathcal{T}_\ell)$ associated with $\mathcal{T}_\ell$, is tensorised with the polynomial space $\mathcal{P}_{\mathcal{P}_\ell}$. The unique Galerkin solution $u_\ell \in V_\ell := V^{(\ell)}_{X\mathcal{P}} = X_\ell \otimes \mathcal{P}_{\mathcal{P}_\ell}$ satisfying (4.36) is computed by solving the associated linear system (4.44) by the SOLVE subroutine,

$$u_\ell = \mathsf{SOLVE}\,(\mathcal{T}_\ell, \mathcal{P}_\ell, a, f)\,,$$

where a and f are the problem data (see (4.5) and (4.8)). In order to control the error in the Galerkin solution u_ℓ, local spatial estimates $\{\eta_{Y\mathcal{P}}(T)\}_{T\in\mathcal{T}_\ell}$ and individual parametric estimates $\{\eta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}$ defined in (5.37) and (5.38), respectively, are computed as described in Section 5.3.1

by the subroutine ESTIMATE:

$$\Big[\{\eta_{Y\mathcal{P}}(T)\}_{T\in\mathcal{T}_\ell},\ \{\eta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}\Big] = \mathsf{ESTIMATE}\Big(u_\ell, \mathcal{T}_\ell, \mathcal{P}_\ell, \mathcal{Q}_\ell, \mathsf{a}, f\Big).$$

Here, the detail index set $\mathcal{Q}_\ell \subset \mathcal{I} \setminus \mathcal{P}_\ell$ is built via (5.35). The total energy norm error estimate $\eta_\ell \approx \eta_{X\mathcal{P}}^{(\ell)}$ is then computed via (5.39). If a prescribed tolerance tol is met, i.e., if $\eta_\ell \leq \mathsf{tol}$, then the adaptive process stops. Otherwise, a more accurate Galerkin solution belonging to an enriched finite-dimensional subspace $V_{\ell+1} \supseteq V_\ell$ needs to be computed. At this stage of the adaptive loop, a marking criterion (see below) returns two subsets, $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ and $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$ where, indeed, only one of them is non-empty according to the type of enrichment that has to be pursued. The selection of such subsets is performed by the same MARK subroutine

$$\mathcal{M}_\ell = \mathsf{MARK}\Big(\{\eta_{Y\mathcal{P}}(T)\}_{T\in\mathcal{T}_\ell}, \theta_X\Big) \qquad \text{and} \qquad \mathcal{M}_\ell = \mathsf{MARK}\Big(\{\eta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}, \theta_{\mathcal{P}}\Big), \tag{5.41}$$

which implements the Dörfler strategy (see Strategy 2.2). Finally, the enriched space $V_{\ell+1}$ is constructed by setting

$$\mathcal{P}_{\ell+1} := \mathcal{P}_\ell \cup \mathcal{M}_\ell \quad \text{and} \quad \mathcal{T}_{\ell+1} = \mathsf{REFINE}(\mathcal{T}_\ell, \mathcal{M}_\ell),$$

where the REFINE subroutine implements NVB refinements (see Section 2.3.2). Note that since either $\mathcal{M}_\ell$ or $\mathcal{M}_\ell$ is empty, only either a parametric enrichment or a mesh-refinement of current triangulation $\mathcal{T}_\ell$ is performed. The algorithm returns a sequence $(\mathcal{T}_\ell)_{\ell\in\mathbb{N}_0}$ of adaptively conforming refined triangulations associated with nested finite elements spaces $(X_\ell)_{\ell\in\mathbb{N}_0}$, i.e., $X_\ell \subseteq X_{\ell+1} \subset H_0^1(D)$ and a sequence $(\mathcal{P}_\ell)_{\ell\in\mathbb{N}_0}$ of adaptively enriched nested index sets, i.e., $\mathcal{P}_\ell \subseteq \mathcal{P}_{\ell+1} \subset \mathcal{I}$.

Let us now describe how the algorithm chooses between mesh-refinements and parametric enrichments of the approximation space. In this respect, we distinguish two possible marking criteria that can be used by the adaptive algorithm.

Marking criterion based on total estimates

This is the criterion listed in Criterion 5.1. Here, we consider the total spatial $\eta_{Y\mathcal{P}}(\mathcal{T}_\ell)$ and total parametric $\eta_{X\mathcal{Q}}(\mathcal{Q}_\ell)$ error estimates given by (5.37) and (5.38), respectively, as follows. If $\eta_{X\mathcal{P}}(\mathcal{T}_\ell)$ is bigger than $\eta_{X\mathcal{Q}}(\mathcal{Q}_\ell)$, the criterion selects a subset $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ of marked elements using the MARK subroutine and set $\mathcal{M}_\ell := \emptyset$. Otherwise (i.e., if $\eta_{X\mathcal{P}}(\mathcal{T}_\ell) < \eta_{X\mathcal{Q}}(\mathcal{Q}_\ell)$), a subset $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$ of marking indices is selected using the MARK subroutine and the criterion sets $\mathcal{M}_\ell := \emptyset$. Note that, in the former case, only a new finite element space is going to be constructed on the next iteration,

Marking criterion for adaptive SGFEM

Input: *error estimates* $\{\eta_{Y\mathcal{P}}(T)\}_{T\in\mathcal{T}_\ell}$, $\{\eta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}$, *and marking parameters* $\theta_X, \theta_{\mathcal{P}} \in (0,1]$.
IF $\eta_{Y\mathcal{P}}(\mathcal{T}_\ell) \geq \eta_{X\mathcal{Q}}(\mathcal{Q}_\ell)$

set $\mathcal{M}_\ell := \mathsf{MARK}\big(\{\eta_{Y\mathcal{P}}(T)\}_{T\in\mathcal{T}_\ell}, \theta_X\big)$ *and* $\mathcal{M}_\ell := \emptyset$;

ELSE

set $\mathcal{M}_\ell := \mathsf{MARK}\big(\{\eta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}, \theta_{\mathcal{P}}\big)$ *and* $\mathcal{M}_\ell := \emptyset$.

END
Output: $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ *and* $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$, *where one of the two subsets is empty.*

Criterion 5.1. Marking criterion for an adaptive SGFEM algorithm driven by hierarchical error estimates.

whereas, in the latter case, the enhanced space is constructed only by parametric enrichment.

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 [54, 55] and [56], where residual-based and local equilibration error estimators were employed, respectively. On the other hand, it was used in the adaptive algorithm with uniform mesh-refinements presented in [29]. Note that, however, the estimate $\eta_{Y\mathcal{P}}(\mathcal{T}_\ell)$ combining all elementwise contributions $\{\eta_{Y\mathcal{P}}(T)\}_{T\in\mathcal{T}_\ell}$ does not necessarily provide an effective estimate of the error reduction that would be achieved if only the marked elements $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ were refined. Likewise, $\eta_{X\mathcal{Q}}(\mathcal{Q}_\ell)$ does not necessarily estimate the error reduction that would be achieved by adding only the subset of marked indices $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$ to $\mathcal{P}_\ell$. These observations motivate the second marking criterion.

Marking criterion based on error reduction estimates

This is the criterion listed in Criterion 5.2. Before deciding on the type of enrichment, the criterion uses the subroutine MARK to select a subset $\widetilde{\mathcal{M}}_\ell \subseteq \mathcal{T}_\ell$ of marked elements as well as a subset $\widetilde{\mathcal{M}}_\ell \subseteq \mathcal{Q}_\ell$ of marked indices. A post-processing step also returns the set $\widetilde{\mathcal{R}}_\ell = \mathcal{T}_\ell \setminus \mathcal{T}_{\ell+1} \subseteq \mathcal{T}_\ell$ of the *refined* elements, i.e., the set consisting of $\widetilde{\mathcal{M}}_\ell$ and all those extra elements that are also marked for refinement by completion steps of the NVB rule (see Figure 2.2). Note that in finding $\widetilde{\mathcal{R}}_\ell$, no mesh-refinement is actually performed: the REFINE subroutine comes with an edge-based procedure which identifies, for a given input set of marked elements, *all* edges of $\mathcal{T}_\ell$ that should be bisected to keep the conformity of the newly refined triangulation before effectively performing the mesh-refinement.

Marking criterion for adaptive SGFEM

Input: *error estimates* $\{\eta_{Y\mathcal{P}}(T)\}_{T\in\mathcal{T}_\ell}$, $\{\eta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}$, *and marking parameters* $\theta_X, \theta_{\mathcal{P}} \in (0,1]$.
DO

set $\widetilde{\mathcal{M}}_\ell := \mathsf{MARK}\big(\{\eta_{Y\mathcal{P}}(T)\}_{T\in\mathcal{T}_\ell}, \theta_X\big)$;

set $\widetilde{\mathcal{R}}_\ell := \mathcal{T}_\ell \setminus \mathcal{T}_{\ell+1}$, *where* $\mathcal{T}_{\ell+1}$ *is defined by* $\mathcal{T}_{\ell+1} := \mathsf{REFINE}(\mathcal{T}_\ell, \widetilde{\mathcal{M}}_\ell)$;

set $\widetilde{\mathcal{M}}_\ell := \mathsf{MARK}\big(\{\eta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}, \theta_{\mathcal{P}}\big)$;

END
IF $\eta_{Y\mathcal{P}}(\widetilde{\mathcal{R}}_\ell) \geq \eta_{X\mathcal{Q}}(\widetilde{\mathcal{M}}_\ell)$

set $\mathcal{M}_\ell := \widetilde{\mathcal{M}}_\ell$ *and* $\mathcal{M}_\ell := \emptyset$;

ELSE

set $\mathcal{M}_\ell := \widetilde{\mathcal{M}}_\ell$ *and* $\mathcal{M}_\ell := \emptyset$;

END
Output: $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ *and* $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$, *where one of the two subsets is empty.*

Criterion 5.2. Marking criterion for an adaptive SGFEM algorithm driven by hierarchical error estimates.

Next, the criterion considers the two error estimates $\eta_{Y\mathcal{P}}(\widetilde{\mathcal{R}}_\ell)$ and $\eta_{X\mathcal{Q}}(\widetilde{\mathcal{M}}_\ell)$ (here, $\eta_{Y\mathcal{P}}(\widetilde{\mathcal{R}}_\ell)$ is defined in obvious way as in (5.40)). Since the sum in $\eta_{Y\mathcal{P}}(\widetilde{\mathcal{R}}_\ell)$ is only over the elements to be refined (resp. the sum in $\eta_{X\mathcal{Q}}(\widetilde{\mathcal{M}}_\ell)$ is over the marked indices to be added to the current index set), the quantity $\eta_{Y\mathcal{P}}(\widetilde{\mathcal{R}}_\ell)$ (resp. $\eta_{X\mathcal{Q}}(\widetilde{\mathcal{M}}_\ell)$) does provide an effective estimate of the error reduction that would be achieved as result of the mesh-refinement (resp. parametric enrichment); cf. (5.30) and (5.31). Therefore, in the spirit of algorithms driven by dominant error reduction estimates, the enrichment type in Criterion 5.2 is chosen by comparing the quantities $\eta_{Y\mathcal{P}}(\widetilde{\mathcal{R}}_\ell)$ and $\eta_{X\mathcal{Q}}(\widetilde{\mathcal{M}}_\ell)$. More precisely, if $\eta_{Y\mathcal{P}}(\widetilde{\mathcal{R}}_\ell) \geq \eta_{X\mathcal{Q}}(\widetilde{\mathcal{M}}_\ell)$, then $\mathcal{M}_\ell$ is set equal to $\widetilde{\mathcal{M}}_\ell$ and $\mathcal{M}_\ell := \emptyset$. Otherwise, $\mathcal{M}_\ell$ is set equal $\widetilde{\mathcal{M}}_\ell$ and $\mathcal{M}_\ell := \emptyset$.

The complete adaptive SGFEM algorithm incorporating both Criteria 5.1 and 5.2 is listed in Algorithm 5.1.

5.4 Numerical experiments

We report the results of running adaptive Algorithm 5.1 for parametric model problem (4.5). These results illustrate some aspects of the design of the algorithm and demonstrate the performance of the two versions using marking Criteria 5.1 and 5.2 described in Section 5.3.3. The numerical experiments are performed using the open source Matlab toolbox Stochastic T-IFISS [28],

Adaptive SGFEM algorithm

Input: *data* a, f; *triangulation* $\mathcal{T}_0$, *index set* $\mathcal{P}_0$; *marking parameters* $\theta_X, \theta_{\mathcal{P}} \in (0,1]$; *tolerance* tol.
FOR $\ell = 0, 1, 2, \ldots$ DO

$u_\ell = \mathsf{SOLVE}(\mathcal{T}_\ell, \mathcal{P}_\ell, a, f)$;

$\Big[\{\eta_{Y\mathcal{P}}(T)\}_{T\in\mathcal{T}_\ell},\ \{\eta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}\Big] = \mathsf{ESTIMATE}\Big(u_\ell, \mathcal{T}_\ell, \mathcal{P}_\ell, \mathcal{Q}_\ell, a, f\Big)$;

$\eta_\ell = \Big(\eta_{Y\mathcal{P}}(\mathcal{T}_\ell)^2 + \eta_{X\mathcal{Q}}(\mathcal{Q}_\ell)^2\Big)^{1/2}$;

IF $\eta_\ell \leq$ tol THEN BREAK; END

select the subsets $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ *and* $\mathcal{M}_\ell \subseteq \mathcal{Q}$ *by using either Criterion 5.1 or Criterion 5.2;*

set $\mathcal{P}_{\ell+1} = \mathcal{P}_\ell \cup \mathcal{M}_\ell$ *and* $\mathcal{T}_{\ell+1} = \mathsf{REFINE}(\mathcal{T}_\ell, \mathcal{M}_\ell)$.

END
Output: *sequence of Galerkin solutions* u_ℓ *and energy error estimates* η_ℓ.

Algorithm 5.1. Adaptive SGFEM algorithm driven by hierarchical error estimates for parametric problem (4.5).

on a desktop computer equipped with an Intel Core CPU i5-4590@3.30GHz and 8.00GB of RAM. A brief description of the toolbox can be found in Appendix B.

5.4.1 Setup of the experiments

In all experiments, we run Algorithm 5.1 using the following initial index set

$$\mathcal{P}_0 := \Big\{\nu^{(1)}, \nu^{(2)}\Big\} = \Big\{\mathbf{0}, (1,0,0,\ldots)\Big\}. \tag{5.42}$$

In $\mathcal{P}_0$, only parameter y_1 is active and the initial polynomial space is given by $\mathcal{P}_{\mathcal{P}_0} = \mathsf{span}\{1, P_{\nu^{(2)}}\}$, where $P_{\nu^{(2)}}$ is a polynomial of degree 1 in y_1. For the computation of detail index sets $\mathcal{Q}_\ell$ defined in (5.35), we fix $M_{\mathcal{Q}} = 1$ and, throughout, we do not investigate the action of larger values.

Let $L = L(\text{tol}) \in \mathbb{N}$ be the smallest integer such that $\eta_L \leq$ tol. We will collect the following output data:

- the number of total iterations L of the adaptive algorithm;
- the overall computational time t (in seconds);
- the final energy error estimate η_L;
- the final number of degrees of freedom $N_L := \dim(V_L) = \dim(X_L)\dim(\mathcal{P}_{\mathcal{P}_L}) = N_{X_L\mathcal{P}_L}$;
- the number of elements $\#\mathcal{T}_L$ of last triangulation $\mathcal{T}_L$;

- the number of indices $\#\mathcal{P}_L$ of last index set $\mathcal{P}_L$ as well as the final number of active parameters $M_{\mathcal{P}_L}$;
- the parametric enrichments of the index set.

In order to test the effectiveness of the error estimation strategy, the hierarchical estimates η_ℓ should be compared with the energy norm of the error $e_\ell = u - u_\ell$. Since the true solution is unknown, we replace $\|e_\ell\|_B$ by the energy norm of $u_{\text{ref}} - u_\ell$, where $u_{\text{ref}} \in V_{\text{ref}} := X_{\text{ref}} \otimes \mathcal{P}_{\mathcal{P}_{\text{ref}}}$ is an accurate, reference solution. To compute u_{ref}, we employ second-order finite element approximations over a fine triangulation $\mathcal{T}_{\text{ref}}$ (i.e., $X_{\text{ref}} := \mathcal{S}^2_0(\mathcal{T}_{\text{ref}})$) and use a large index set $\mathcal{P}_{\text{ref}}$ which are both specified in the experiments. Then, we define the following effectivity indices

$$\xi_\ell := \frac{\eta_\ell}{\|u_{\text{ref}} - u_\ell\|_B} = \frac{\eta_\ell}{(\|u_{\text{ref}}\|_B^2 - \|u_\ell\|_B^2)^{1/2}}, \qquad \ell = 0, \dots, L. \tag{5.43}$$

Note that the equality in (5.43) holds due to Galerkin orthogonality and the symmetry of the bilinear form $B(\cdot,\cdot)$.

In what follows, we will write Algorithm 5.1v1 (resp. Algorithm 5.1v2) to refer to the version of adaptive Algorithm 5.1 which uses marking Criterion 5.1 (resp. Criterion 5.2).

5.4.2 Experiment 1 - Spatially regular solution on square domain

In the first experiment, we consider the parametric model problem (4.5) posed on the square domain $D = (0,1)^2$, and we set the right-hand side source $f(\mathbf{x}) := 1$ for all $\mathbf{x} \in D$. Following [54, 55], we fix $a_0(\mathbf{x}) = 1$ for all $\mathbf{x} \in D$ and choose the expansion coefficients $(a_m)_{m\in\mathbb{N}}$ in (4.8) to represent planar Fourier modes of increasing total order, i.e.,

$$a(\mathbf{x},\mathbf{y}) = 1 + \sum_{m=1}^{\infty} \Big[\alpha_m \cos(2\pi\beta_1(m)\,x_1)\cos(2\pi\beta_2(m)\,x_2)\Big] y_m, \quad \mathbf{x} \in D,\ \mathbf{y} \in \Gamma. \tag{5.44}$$

Here, $\alpha_m := Am^{-\sigma}$ represents the amplitude of the coefficients, where $\sigma > 1$ and $0 < A < 1/\zeta(\sigma)$, with ζ being the Riemann zeta function, whereas $\beta_1, \beta_2 : \mathbb{N} \to \mathbb{N}$ are defined by

$$\beta_1(m) := m - k(m)(k(m)+1)/2 \quad \text{and} \quad \beta_2(m) := k(m) - \beta_1(m), \qquad m \in \mathbb{N}$$

with $k(m) := \left\lfloor -1/2 + \sqrt{1/4 + 2m} \right\rfloor$. Note that with this choice of expansion coefficients, the weak formulation (4.17) is well-posed since $a_0^{\min} = a_0^{\max} = 1$ in (4.9) and $\gamma = A\zeta(\sigma) < 1$ as required by (4.10). In particular, by setting $\sigma = 2$, we select A such that $\tau = A\zeta(\sigma) = 0.9$. This choice corresponds to a slow decay of the amplitudes α_m and gives $A \approx 0.547$ (cf. [54, Section 11.1.1]). Furthermore, we assume that parameters $y_m \in \Gamma_m = [-1,1]$ in (5.44) are the images of uniformly

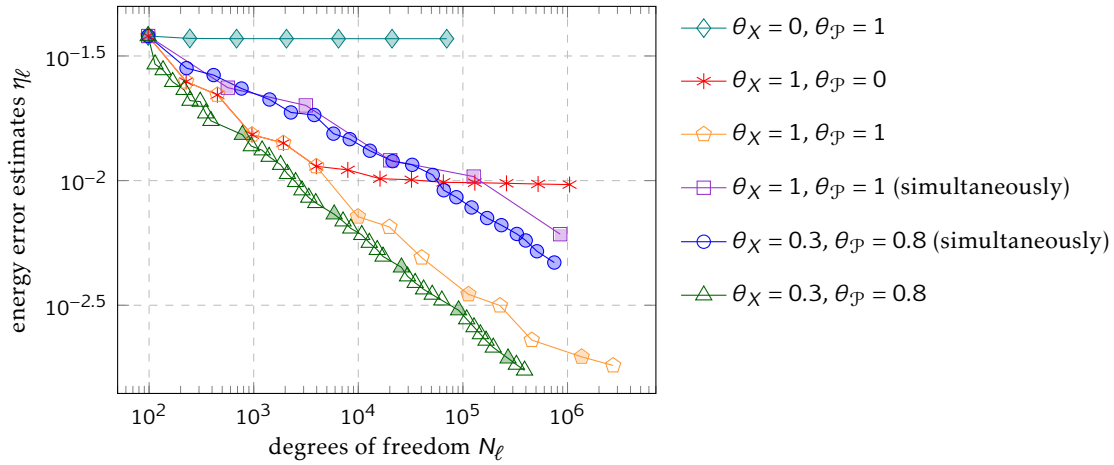


Figure 5.2. Numerical experiment of Section 5.4.2. Decay of energy norm estimates η_ℓ computed at each step of Algorithm 5.1v2 for different sets of marking parameters as well as for the case of Criterion 5.2 modified so that mesh-refinements and parametric enrichments are enforced simultaneously at each iteration. Filled markers indicate iterations at which parametric enrichments occur.

distributed independent mean-zero random variables. In this case, $d\pi_m = dy_m/2$ for all $m \in \mathbb{N}$ and the orthonormal polynomials basis of $L^2_{\pi_m}(\Gamma_m)$ consists of Legende polynomials (see Example 4.2.1). The same model problem as described above has been used in numerical experiments in [54, 55, 29, 56].

The first aim in this experiment is to show the advantages of using adaptivity in *both* components of Galerkin approximations. To this end, starting from the initial coarse triangulation $\mathcal{T}_0$ depicted in Figure 5.3(a), we run Algorithm 5.1v2 with four different sets of marking parameters. In addition, we consider a modification of Algorithm 5.1v2, in which Criterion 5.2 is amended so that both mesh-refinement and parametric enrichment are enforced at each iteration. We plot the computed error estimates η_ℓ in Figure 5.2.

In the cases where only one component of the Galerkin approximation is enriched (i.e., if either $\theta_\mathcal{X}$ or $\theta_\mathcal{P}$ is equal to 0) the error estimates η_ℓ quickly stagnate as iterations progress. If both components are enriched but no adaptivity is used (i.e., for $\theta_\mathcal{X} = \theta_\mathcal{P} = 1$), then η_ℓ decay throughout all iterations. However, in this case, the overall decay rate deteriorates due to the number of degrees of freedom growing fast, in particular, during the iterations where parametric enrichments occur (see the filled pentagon markers in Figure 5.2). An even greater deterioration of the decay rate is also observed for the case in which both components are fully enriched ($\theta_\mathcal{X} = \theta_\mathcal{P} = 1$) simultaneously at each iteration. On the other hand, a similar decay is also obtained if we enforce both enhancements at each iteration but marking parameters less than one are used ($\theta = 0.3$ and

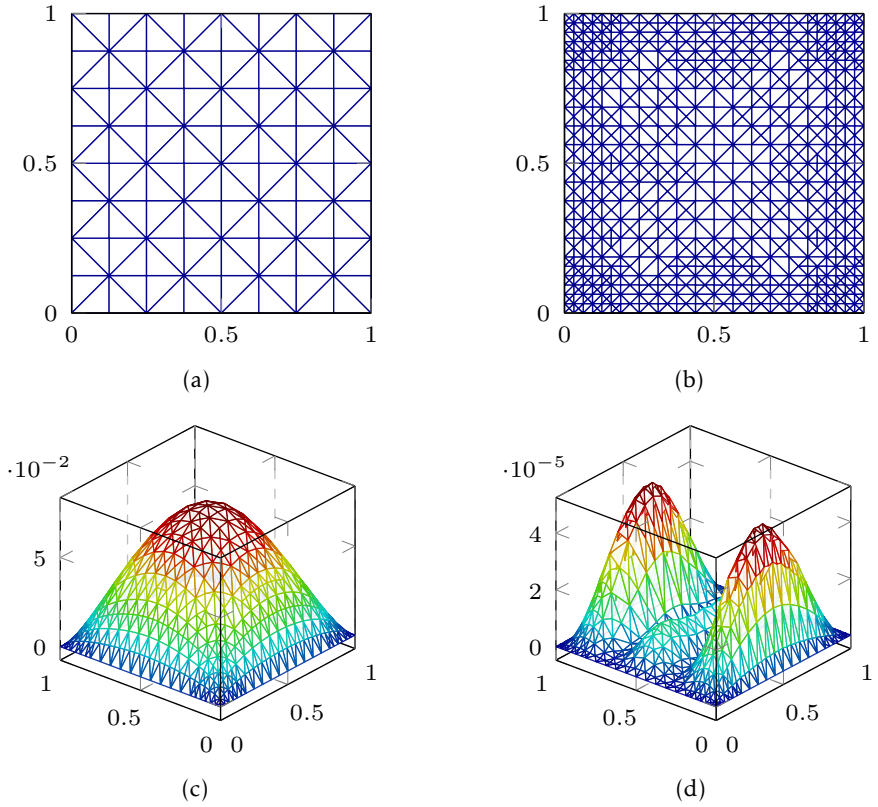


Figure 5.3. Numerical experiment of Section 5.4.2. (a) Initial coarse triangulation $\mathcal{T}_0$; (b) Adaptively refined triangulation produced by Algorithm 5.1v1; (c)–(d) The mean field $\mathbb{E}[u_{X\mathcal{P}}]$ and the variance $\mathsf{Var}(u_{X\mathcal{P}})$ of the computed SGFEM solution, respectively.

$\theta_{\mathcal{P}} = 0.8$). Finally, we observe that the best decay rate is obtained when adaptivity is used for both components of Galerkin approximations in Criterion 5.2 with parameters $\theta = 0.3$ and $\theta_{\mathcal{P}} = 0.8$ (see triangle markers in Figure 5.2). Clearly, adaptive enrichment in *both* components provides more balanced approximations with less degrees of freedom and leads to faster convergence rates.

Let us now run both Algorithms 5.1v1 and 5.1v2 with the following sets of input parameters. For marking purposes, we use two sets of threshold parameters, (i) $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$ and (ii) $\theta_X = 0.2$, $\theta_{\mathcal{P}} = 0.9$. The stopping tolerance tol = 1.5e-3 is set in all cases. The results of these computations are presented in Table 5.1 and in Figures 5.3, 5.4, and 5.5.

Figure 5.3(b) shows the locally refined triangulation produced by Algorithm 5.1v1 in case (i) when an intermediate tolerance was met (similar triangulations were produced in all other cases). Figures 5.3(c) and 5.3(d) show the mean and the variance of the computed SGFEM solution, respectively (see (4.48)). 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 mainly

	$\theta_X = 0.5,\ \theta_{\mathcal{P}} = 0.9$				$\theta_X = 0.2,\ \theta_{\mathcal{P}} = 0.9$			
	Algorithm 5.1v1		Algorithm 5.1v2		Algorithm 5.1v1		Algorithm 5.1v2	
L	27		26		64		61	
t (sec)	254		179		474		391	
η_L	1.2652e-03		1.4438e-03		1.2509e-03		1.4369e-03	
N_L	997,763		748,558		986,769		730,319	
$\#\mathcal{T}_L$	87,520		65,750		86,552		64,156	
$\#\mathcal{P}_L$	23		23		23		23	
$M_{\mathcal{P}_L}$	6		6		6		6	
$\mathcal{P}_\ell$	$\ell = 9$	(0 1) (2 0)	$\ell = 8$	(0 1) (2 0)	$\ell = 21$	(0 1) (2 0)	$\ell = 10$	(0 1) (2 0)
	$\ell = 15$	(0 0 1) (1 1 0) (3 0 0)	$\ell = 13$	(0 0 1) (1 1 0) (3 0 0)	$\ell = 35$	(0 0 1) (1 1 0) (3 0 0)	$\ell = 23$	(0 0 1) (1 1 0) (3 0 0)
	$\ell = 20$	(0 0 0 1) (1 0 1 0) (2 1 0 0)	$\ell = 19$	(0 0 0 1) (1 0 1 0) (2 1 0 0)	$\ell = 48$	(0 0 0 1) (1 0 1 0) (2 1 0 0)	$\ell = 36$	(0 0 0 1) (1 0 1 0) (2 1 0 0)
	$\ell = 24$	(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)	$\ell = 22$	(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)	$\ell = 56$	(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)	$\ell = 44$	(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)
	$\ell = 27$	(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)	$\ell = 26$	(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)	$\ell = 64$	(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)	$\ell = 51$	(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 5.1. The results of running Algorithms 5.1v1 and 5.1v2 with two sets of marking parameters for the model problem in Section 5.4.2.

refined towards the corners of the domain.

The results in Table 5.1 evidence some differences in the performance of the algorithm using Criteria 5.1 and 5.2 in terms of computational times, final number of elements, and total number of degrees of freedom (cf. the values of t, $\#\mathcal{T}_L$, and N_L in Table 5.1). In particular, Algorithm 5.1v2 took less iterations and reached the tolerance faster than Algorithm 5.1v1 (e.g., about 33% of time saved in case (ii)). However, in cases (i) and (ii), both Algorithms 5.1v1 and 5.1v2 produced the same final index set $\mathcal{P}_L$ with 23 indices corresponding to polynomials of total degree 4 in 6 active parameters. We also note that by design, the use of Criterion 5.2 triggers polynomial enrichments at earlier iterations than the case when Criterion 5.1 is used. This results in a balanced refinement of spatial and parametric components of Galerkin approximations generated by running Algorithm 5.1v2 and this is one of the reasons why it is faster and overall more efficient than

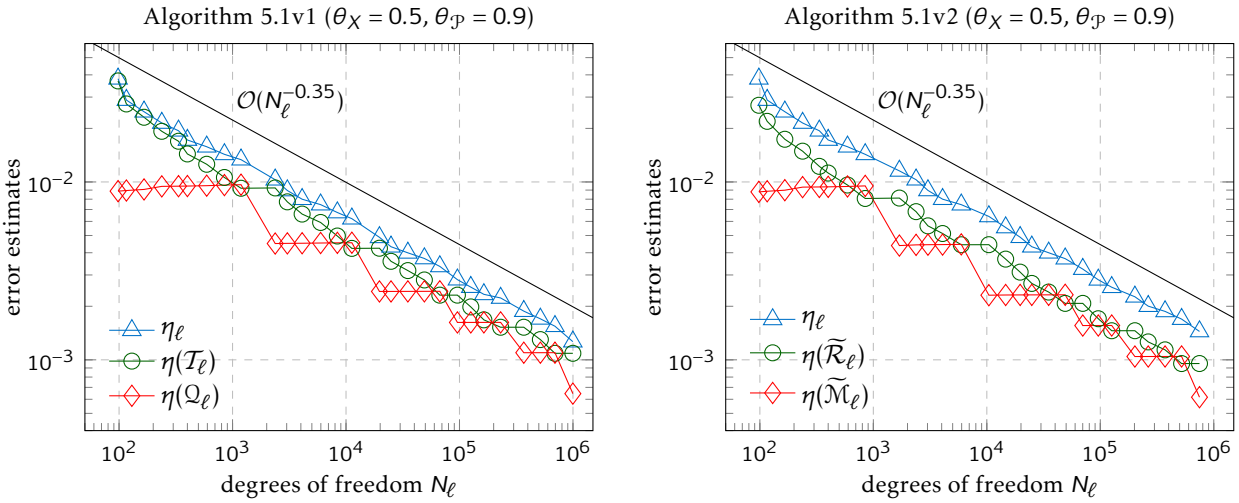


Figure 5.4. Total and local error estimates at each step of Algorithms 5.1v1 (left) and 5.1v2 (right) with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$ (case (i)) for the model problem in Section 5.4.2.

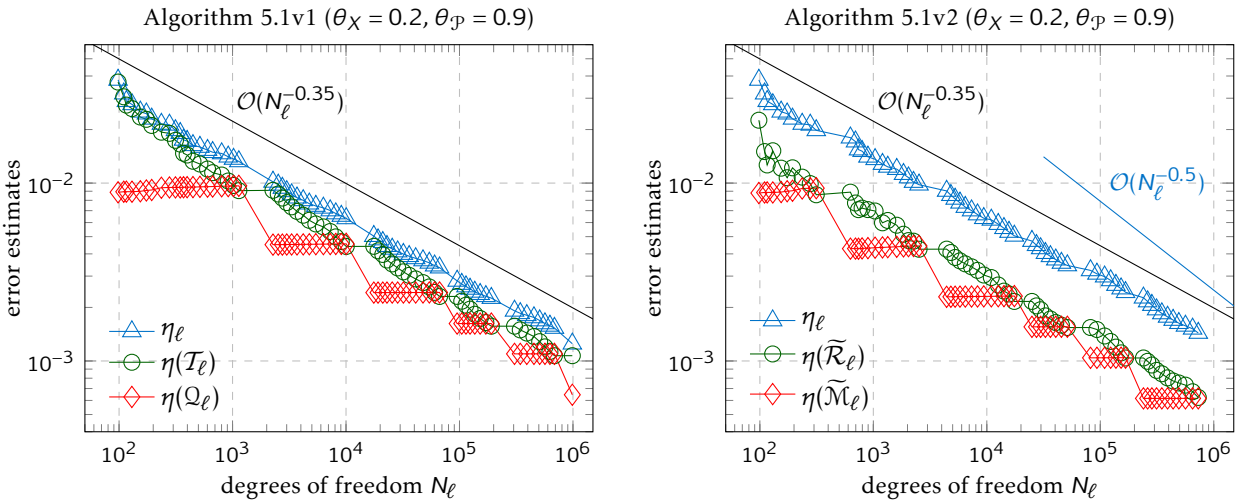


Figure 5.5. Total and local error estimates at each step of Algorithms 5.1v1 (left) and 5.1v2 (right) with $\theta_X = 0.2$, $\theta_{\mathcal{P}} = 0.9$ (case (ii)) for the model problem in Section 5.4.2.

Algorithm 5.1v1 in this experiment.

By looking now at Figures 5.4 and 5.5, we observe that the total error estimates η_ℓ decay with an overall rate of about $\mathcal{O}(N^{-0.35})$ for both Algorithms 5.1v1 and 5.1v2 and both sets of marking parameters. However, due to spatial regularity of the solution, one may expect the error estimates to decay with the optimal rate $\mathcal{O}(N^{-1/2})$ during mesh-refinement steps, cf. [55] (mesh-refinements can be identified on the graphs as the steps where the estimates $\eta(\mathcal{T}_\ell)$ and $\eta(\widetilde{\mathcal{R}}_\ell)$ decay). It turns out that Algorithm 5.1 does not achieve this optimal decay rate during mesh-refinement stages in case (i) ($\theta_X = 0.5$); see Figure 5.4 (also cf. [55, Figure 1] in the case of $\theta_X = 0.4$). However, in case (ii) ($\theta_X = 0.2$), Figure 5.5 shows that the decay rate during spatial refinement steps is very

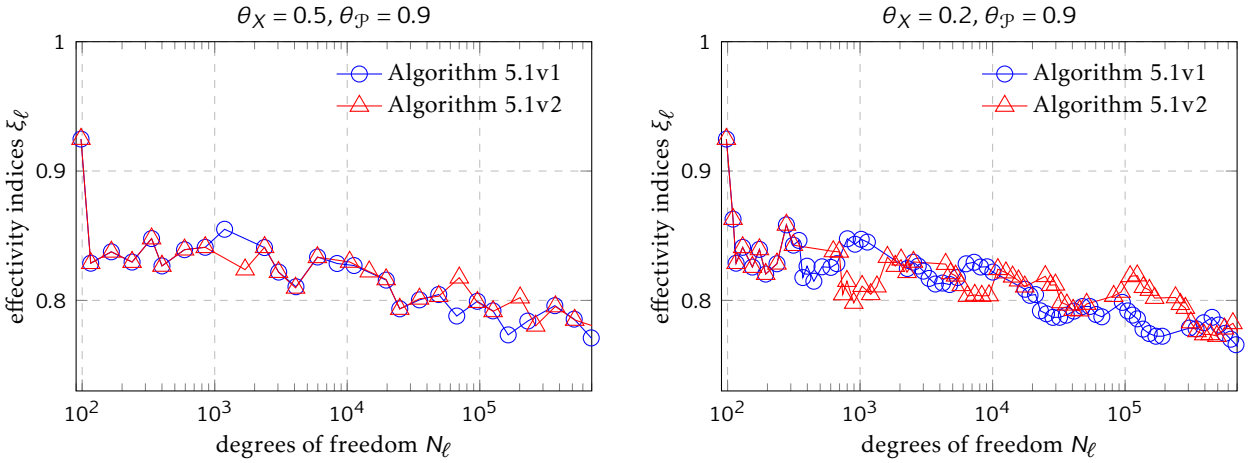


Figure 5.6. The effectivity indices for the SGFEM solutions of the model problem in Section 5.4.2 computed by Algorithms 5.1v1 and 5.1v2 with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$ (case (i), left) and $\theta_X = 0.2$, $\theta_{\mathcal{P}} = 0.9$ (case (ii), right).

close to the optimal one when Criterion 5.2 is used, whereas it is still far from being optimal if Criterion 5.1 is used by the algorithm. We also note that, since polynomial enrichments are triggered earlier by Criterion 5.2, 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 Algorithm 5.1v1.

Finally, in both cases (i) and (ii), we compute the effectivity indices (5.43). In this case, we employ the reference Galerkin solution u_{ref} from [29, Section 6] with corresponding energy norm $\|u_{\text{ref}}\|_B = 1.90117\text{e-}01$. The effectivity indices for both cases (i) and (ii) and both Algorithms 5.1v1 and 5.1v2 are plotted in Figure 5.6. We can see that the they are less than unity throughout all iterations and tend to be close to 0.8 as iterations progress.

Based on the obtained results, we conclude that Algorithm 5.1v2 is more efficient than Algorithm 5.1v1 for the considered parametric problem on the square domain. Indeed, Algorithm 5.1v2 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 parameter θ_X is sufficiently small (e.g., $\theta_X = 0.2$). 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.

5.4.3 Experiment 2 - Spatially singular solution on L-shaped domain

In the second experiment, we compare the performance of the two versions of Algorithm 5.1 for the same parametric model problem described in Section 5.4.2 but now posed on the L-shaped

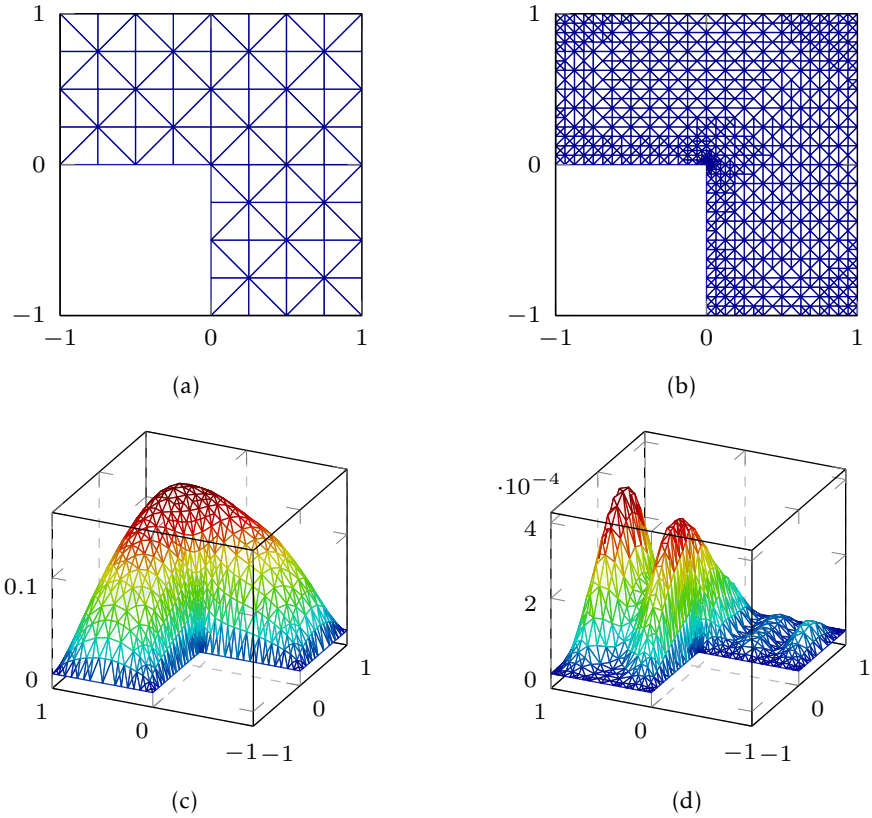


Figure 5.7. Numerical experiment of Section 5.4.3. (a) Initial coarse triangulation $\mathcal{T}_0$; (b) Adaptively refined triangulation produced by Algorithm 5.1v1; (c)–(d) The mean field $\mathbb{E}[u_{X\mathcal{P}}]$ and the variance $\mathsf{Var}(u_{X\mathcal{P}})$ of the computed SGFEM solution.

domain $D = (-1,1)^2 \setminus (-1,0]^2$. Exactly the same parametric problem has been solved numerically in [54, 55, 56].

We use the initial coarse triangulation $\mathcal{T}_0$ depicted in Figure 5.7(a). Similarly to the experiment in Section 5.4.2, for marking purposes we use two sets of parameters, (i) $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.8$ and (ii) $\theta_X = 0.2$, $\theta_{\mathcal{P}} = 0.8$, and the same stopping tolerance tol = 5.0e-3 is set for both cases. The results of these computations are presented in Table 5.2 and in Figures 5.7, 5.8, and 5.9.

Figure 5.7(b) shows the locally refined triangulation produced by Algorithm 5.1v1 in case (ii) when an intermediate tolerance was met (triangulations with a similar patterns were produced in all other cases). Figures 5.7(c) and 5.7(d) show the mean and the variance of the computed SGFEM solution, respectively. Observe that the adaptively refined triangulation effectively identifies the area of singular behaviour 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, the 'roughness' of the variance in

	$\theta_X = 0.5,\ \theta_{\mathcal{P}} = 0.8$				$\theta_X = 0.2,\ \theta_{\mathcal{P}} = 0.8$			
	Algorithm 5.1v1		Algorithm 5.1v2		Algorithm 5.1v1		Algorithm 5.1v2	
L	27		27		65		63	
t (sec)	193		194		360		333	
η_L	4.7170e-03		4.7160e-03		4.8340e-03		4.9659e-03	
N_L	664,729		665,366		576,121		603,594	
$\#\mathcal{T}_L$	103,206		103,304		89,480		67,770	
$\#\mathcal{P}_L$	13		13		13		18	
$M_{\mathcal{P}_L}$	5		5		5		6	
$\mathcal{P}_\ell$	$\ell = 12$	(0 1) (2 0)	$\ell = 11$	(0 1) (2 0)	$\ell = 29$	(0 1) (2 0)	$\ell = 20$	(0 1) (2 0)
	$\ell = 19$	(0 0 1) (1 1 0)	$\ell = 17$	(0 0 1) (1 1 0)	$\ell = 45$	(0 0 1) (1 1 0)	$\ell = 35$	(0 0 1) (1 1 0)
	$\ell = 22$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$\ell = 21$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$\ell = 53$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$\ell = 43$	(0 0 0 1) (1 0 1 0) (3 0 0 0)
	$\ell = 26$	(0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (2 1 0 0 0)	$\ell = 25$	(0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (2 1 0 0 0)	$\ell = 62$	(0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (2 1 0 0 0)	$\ell = 52$	(0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (2 1 0 0 0)
							$\ell = 60$	(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 5.2. The results of running Algorithms 5.1v1 and 5.1v2 with two sets of marking parameters for the model problem in Section 5.4.3.

some parts of the domain does not have a significant impact on mesh-refinements in those areas.

Table 5.2 shows the final outputs of all computations in this experiment. By looking at the results for case (i), we do not observe significant differences between the approximations produced by the algorithm using Criteria 5.1 and 5.2. Indeed, the tolerance was reached after the same number of iterations ($L = 27$) and the same final index set (with $\#\mathcal{P}_L = 13$ indices), and the number of elements in final triangulations was comparable. Also, both Algorithms 5.1v1 and 5.1v2 took nearly the same time to reach the tolerance.

In case (ii), the differences are more evident. To start with, Algorithm 5.1v1 needed two iterations more than Algorithm 5.1v2 to reach the tolerance. Furthermore, it produced a more refined triangulation than Algorithm 5.1v2 did (cf. the values of $\#\mathcal{T}_L$ in Table 5.2 in case (ii)). On the other hand, Algorithm 5.1v2 generated a more developed index set (with $\#\mathcal{P}_L = 18$ indices) with more active parameters and higher degree of polynomial approximation in these parameters. This explains why Algorithm 5.1v2 terminated with a slightly bigger number of total degrees of freedom

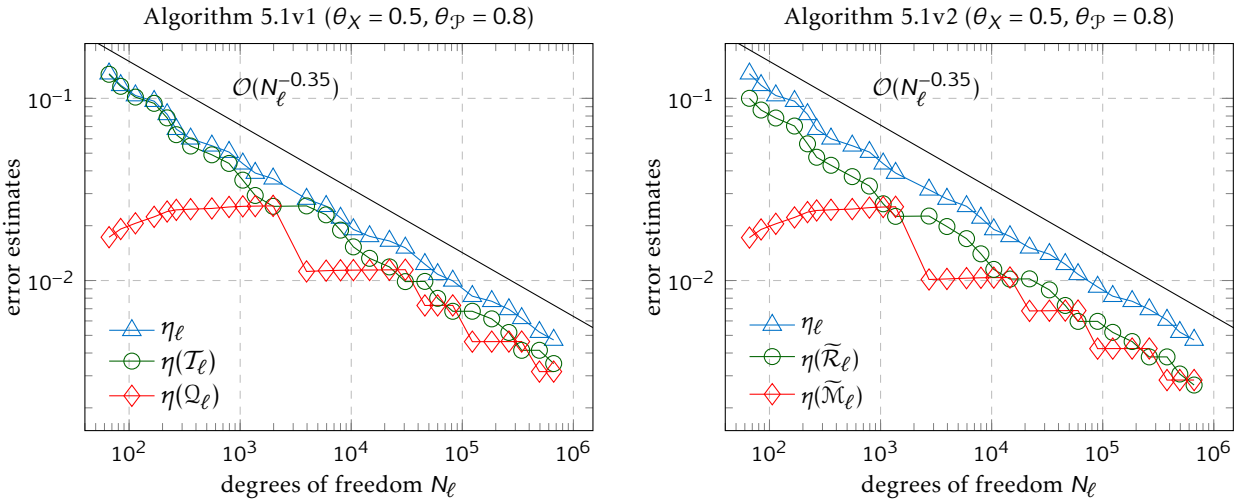


Figure 5.8. Total and local error estimates at each step of Algorithms 5.1v1 (left) and 5.1v2 (right) with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.8$ (case (i)) for the model problem in Section 5.4.3.

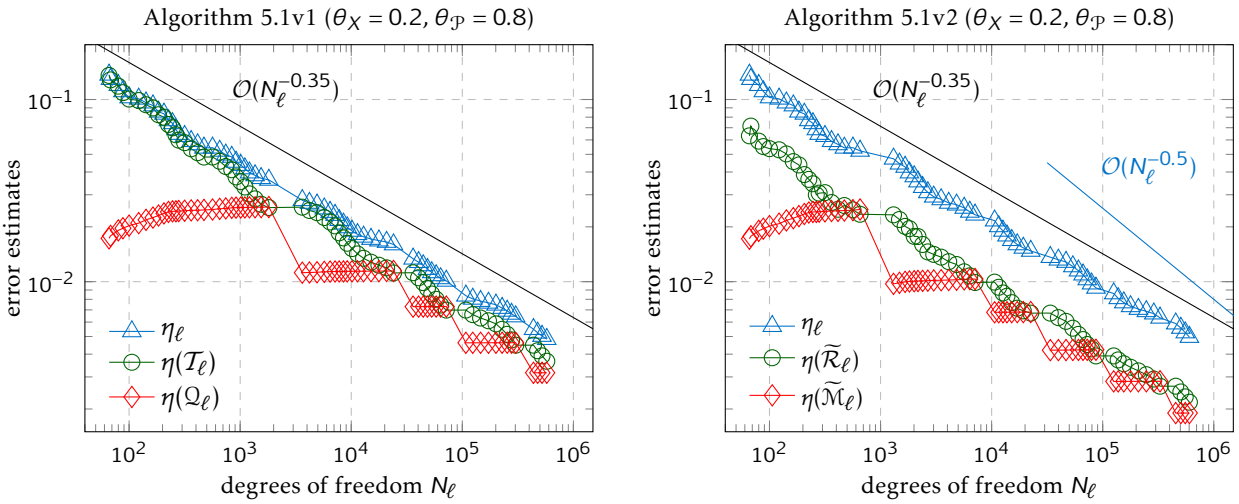


Figure 5.9. Total and local error estimates at each step of Algorithms 5.1v1 (left) and 5.1v2 (right) with $\theta_X = 0.2$, $\theta_{\mathcal{P}} = 0.8$ (case (ii)) for the model problem in Section 5.4.3.

N_L in this case; nevertheless, Algorithm 5.1v2 was 7.5% faster than Algorithm 5.1v1. As already observed in the experiment of Section 5.4.2, this was due to polynomial enrichments triggered at earlier iterations.

By looking now at Figures 5.8 and 5.9, we see that the overall convergence rate for total error estimates η_ℓ is about $\mathcal{O}(N^{-0.35})$ for both Algorithms 5.1v1 and 5.1v2 and for both sets of marking parameters. Notice that the optimal rate $\mathcal{O}(N^{-1/2})$ is not achieved in case (i) due to the fact that the marking parameter $\theta_X = 0.5$ is not sufficiently small (see, e.g., [31, 52]). 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 Algorithm 5.1v2. This observation is consistent with the one made in the experiment of Section 5.4.2

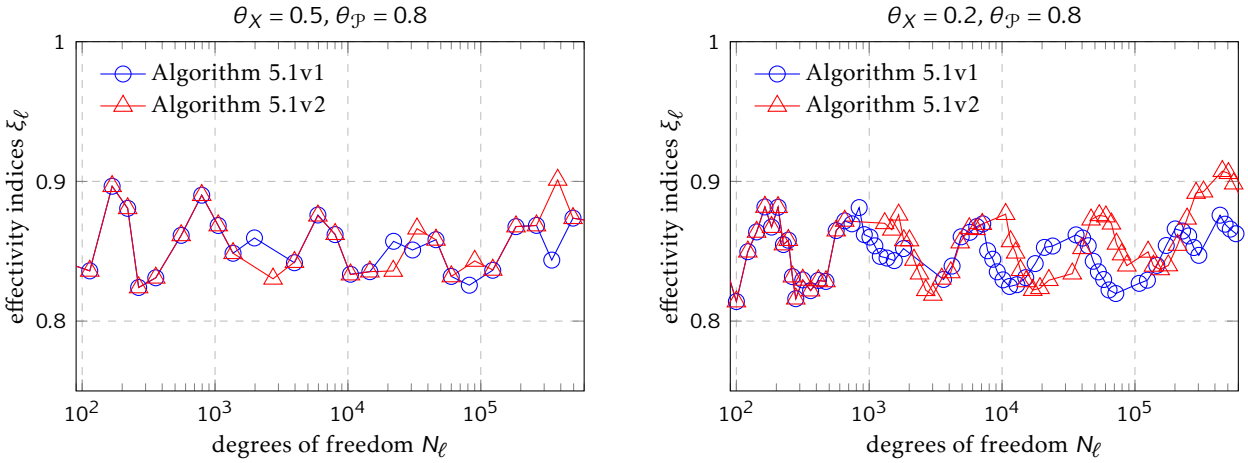


Figure 5.10. The effectivity indices for the SGFEM solutions of the model problem in Section 5.4.3 computed by Algorithms 5.1v1 and 5.1v2 with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.8$ (case (i), left) and $\theta_X = 0.2$, $\theta_{\mathcal{P}} = 0.8$ (case (ii), right). The energy norm of the associated reference solution is $\|u_{\text{ref}}\|_B = 4.701397\text{e-}01$.

on the square domain.

In this experiment, we computed the effectivity indices ξ_ℓ defined in (5.43), by employing a reference Galerkin solution u_{ref} computed over the triangulation $\mathcal{T}_{\text{ref}}$, with $\mathcal{T}_{\text{ref}}$ being the uniform refinement of $\mathcal{T}_L$ produced by Algorithm 5.1v2 in case (i) and using the reference index set $\mathcal{P}_{\text{ref}}$ to be equal to the large index set $\mathcal{P}_L$ generated by Algorithm 5.1v2 in case (ii). The computed effectivity indices are plotted in Figure 5.10. In both cases (i) and (ii), they lie within the interval $(0.8, 0.93)$ throughout all iterations.

In agreement with the results of the experiment in Section 5.4.2, we conclude that for the parametric problem on the L-shaped domain with spatially singular solution, for a sufficiently small marking parameter θ_X (e.g., $\theta_X = 0.2$), Algorithm 5.1 is overall more efficient when using Criterion 5.2. In this case, the algorithm produces more accurate parametric approximations by generating richer index sets, and the associated total error estimates decay with an optimal rate during mesh-refinement steps.

5.4.4 Experiment 3 - Spatially singular solution on slit domain

In this third experiment, we consider the parametric model problem (4.5) posed on the slit domain $D = (-1,1)^2 \setminus ([-1,0] \times \{0\})$. Note that the boundary of this domain is non-Lipschitz (see Figure 5.11(a)); however, the problem on D can be seen as a limit case of the problem on the Lipschitz domain $D_\delta = (-1,1)^2 \setminus \overline{T}_\delta$ as $\delta \to 0$, where $T_\delta = \text{conv}\big(\{(0,0),(-1,\delta),(-1,-\delta)\}\big)$ (cf. [125,

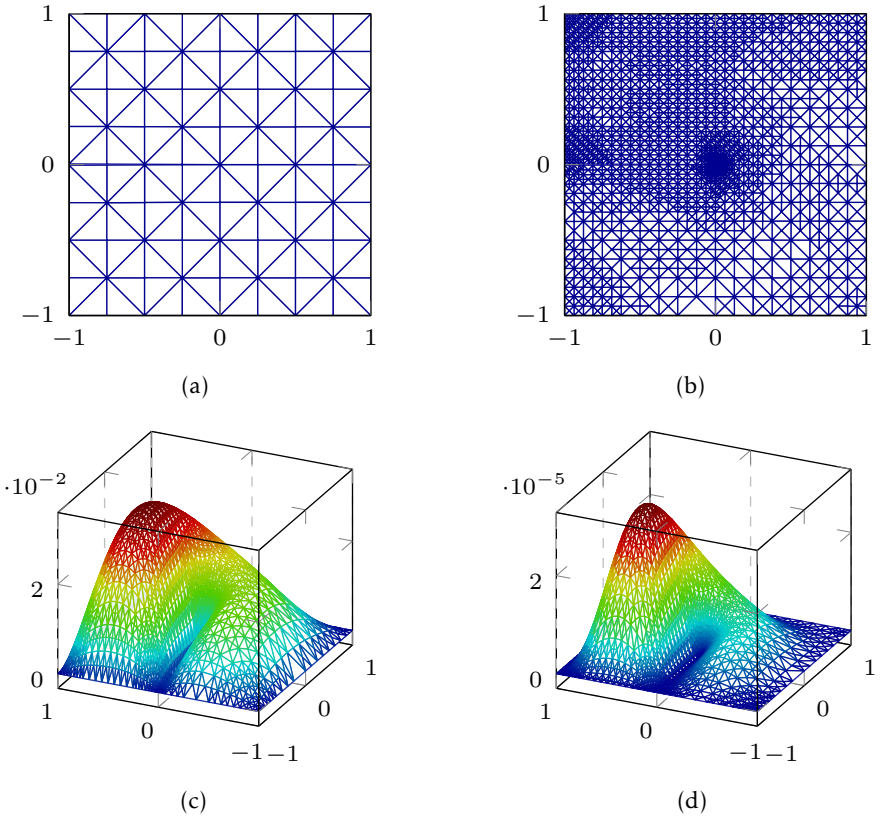


Figure 5.11. Numerical experiment of Section 5.4.4. (a) Initial coarse triangulation $\mathcal{T}_0$; (b) Adaptively refined triangulation produced by Algorithm 5.1v1; (c)–(d) The mean field $\mathbb{E}[u_{X^{\mathcal{P}}}]$ and the variance $\mathrm{Var}(u_{X^{\mathcal{P}}})$ of the computed SGFEM solution.

p. 259])[2]. Therefore, all computations in this experiment were performed for the domain $D = D_\delta$ with $\delta = 0.005$.

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

$$a(\mathbf{x},\mathbf{y}) \;=\; 3 \;+\; \sqrt{3}\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sqrt{\nu_{ij}}\,\phi_{ij}(\mathbf{x})\,y_{ij}\,, \tag{5.45}$$

where $\phi_{00} := 1$, $\nu_{00} := 1/4$, ϕ_{ij} and ν_{ij} are defined in (3.18) for all $i,j \in \mathbb{N}$ (with $\ell = 0.9$), and $y_{ij} \in [-1,1]$ are the images of uniformly distributed independent mean-zero random variables for all $i,j \in \mathbb{N}_0$. Notice that $\sqrt{3}$ in (5.45) is the normalisation constant such that $\mathrm{Var}(\sqrt{3}\,y_{ij}) = 1$; see Example 3.3.2. We rewrite the sum in (5.45) in terms of a single index m in such a way that

[2] We also refer to [78, Section 2.7, p. 83] for a discussion about the well-posedness of the standard weak formulation for the deterministic Poisson problem on the slit domain, in particular, in the case of homogeneous Dirichlet boundary conditions.

	Algorithm 5.1v1					
	$\theta_X = 0.2,\ \ \theta_{\mathcal{P}} = 0.9$		$\theta_X = 0.5,\ \ \theta_{\mathcal{P}} = 0.9$		$\theta_X = 0.35,\ \ \theta_{\mathcal{P}} = 0.65$	
L	71		31		43	
t (sec)	578		290		399	
η_L	1.4808e-03		1.4912e-03		1.4558e-03	
N_L	810,441		852,480		895,617	
$\#\mathcal{T}_L$	181,379		190,720		200,345	
$\#\mathcal{P}_L$	9		9		9	
$M_{\mathcal{P}_L}$	4		4		4	
$\mathcal{P}_\ell$	$\ell = 36$	(0 1) (2 0)	$\ell = 16$	(0 1) (2 0)	$\ell = 21$	(0 1) (2 0)
	$\ell = 39$	(0 0 1) (1 1 0)	$\ell = 18$	(0 0 1) (1 1 0)	$\ell = 23$	(0 0 1)
	$\ell = 55$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$\ell = 25$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$\ell = 31$	(1 0 1) (1 1 0)
					$\ell = 37$	(0 0 0 1) (3 0 0 0)

Table 5.3. The results of running Algorithm 5.1v1 with three sets of marking parameters for the model problem in Section 5.4.4.

corresponding values ν_m appear in descending order of magnitudes, i.e.,

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

For the model problem described above, we run the two versions of Algorithm 5.1 using either Criterion 5.1 or Criterion 5.2 with the initial coarse triangulation $\mathcal{T}_0$ depicted in Figure 5.11(a). Aiming to understand the influence of both marking parameters θ_X and $\theta_{\mathcal{P}}$, we perform computations with three sets of Dörfler marking parameters: (i) $\theta_X = 0.2$, $\theta_{\mathcal{P}} = 0.9$, (ii) $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$, and (iii) $\theta_X = 0.35$, $\theta_{\mathcal{P}} = 0.65$. The same stopping tolerance tol = 1.5e-3 was set in all computations. The results of these computations are presented in Tables 5.3 and 5.4, and in Figures 5.11 and 5.12.

Figure 5.11(b) shows the locally refined triangulation produced by Algorithm 5.1v1 with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$ (case (ii)) when an intermediate tolerance was met. In Figure 5.11(c) and 5.11(d), the mean and the variance of the computed SGFEM solution are plotted. As in previous experiments (cf. Sections 5.4.2 and 5.4.3), we see that the algorithm performs effective adaptive mesh-refinements in the areas where the mean of the solution is 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 5.3 and 5.4, we can see that among six computations car-

	Algorithm 5.1v2					
	$\theta_X = 0.2,\ \ \theta_{\mathcal{P}} = 0.9$		$\theta_X = 0.5,\ \ \theta_{\mathcal{P}} = 0.9$		$\theta_X = 0.35,\ \ \theta_{\mathcal{P}} = 0.65$	
L	66		30		41	
t (sec)	323		289		223	
η_L	1.4997e-03		1.2802e-03		1.4254e-03	
N_L	638,370		1,042,365		671,307	
$\#\mathcal{T}_L$	85,961		140,124		104,258	
$\#\mathcal{P}_L$	15		15		13	
$M_{\mathcal{P}_L}$	4		4		4	
$\mathcal{P}_\ell$	$\ell = 25$	(0 1) (2 0)	$\ell = 14$	(0 1) (2 0)	$\ell = 17$	(0 1) (2 0)
	$\ell = 28$	(0 0 1) (1 1 0)	$\ell = 16$	(0 0 1) (1 1 0)	$\ell = 20$	(0 0 1)
	$\ell = 44$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$\ell = 23$	(0 0 0 1) (1 0 1 0) (3 0 0 0)	$\ell = 28$	(1 0 1) (1 1 0)
	$\ell = 60$	(0 0 2 0) (0 1 1 0) (0 2 0 0) (1 0 0 1) (2 0 1 0) (2 1 0 0)	$\ell = 30$	(0 0 2 0) (0 1 1 0) (0 2 0 0) (1 0 0 1) (2 0 1 0) (2 1 0 0)	$\ell = 34$ $\ell = 41$	(0 0 0 1) (3 0 0 0) (0 1 1 1) (1 0 0 1) (2 0 1 0) (2 1 0 0)

Table 5.4. The results of running Algorithm 5.1v2 with three sets of marking parameters for the model problem in Section 5.4.4.

ried out in this experiment, the best performance in terms of computational time was achieved by Algorithm 5.1v2 with $\theta_X = 0.35$, $\theta_{\mathcal{P}} = 0.65$ (case (iii)). In particular, it was about 44% faster than Algorithm 5.1v1 with the same marking parameters. In agreement with the results of experiments of Sections 5.4.2 and 5.4.3, Algorithm 5.1v2 produced less refined triangulations and triggered polynomial enrichments earlier than Algorithm 5.1v1 in all three cases. Notice that the advantages of using a smaller marking threshold θ_X are again more evident when running Algorithm 5.1v2: final triangulations in cases (i) and (iii) (i.e., for $\theta_X = 0.2$ and $\theta_X = 0.35$) have about twice less elements than the corresponding cases for Algorithm 5.1v1. In addition, in both cases (i) and (ii), Algorithm 5.1v2 needed less iterations than Algorithm 5.1v1 to reach the set tolerance. Also, along with less computational time, Algorithm 5.1v2 produced final index sets $\mathcal{P}_L$ more developed than index sets generated by Algorithm 5.1v1 in all cases (15 versus 9 indices in cases (i) and (ii) and 13 versus 9 in case (iii)). These conclusions are in agreement with numerical results for the parametric problem with spatially singular solution of Section 5.4.3, and they confirm that in terms of efficiency, Algorithm 5.1v2 is more sensitive to over-refined triangulations than Algorithm 5.1v1.

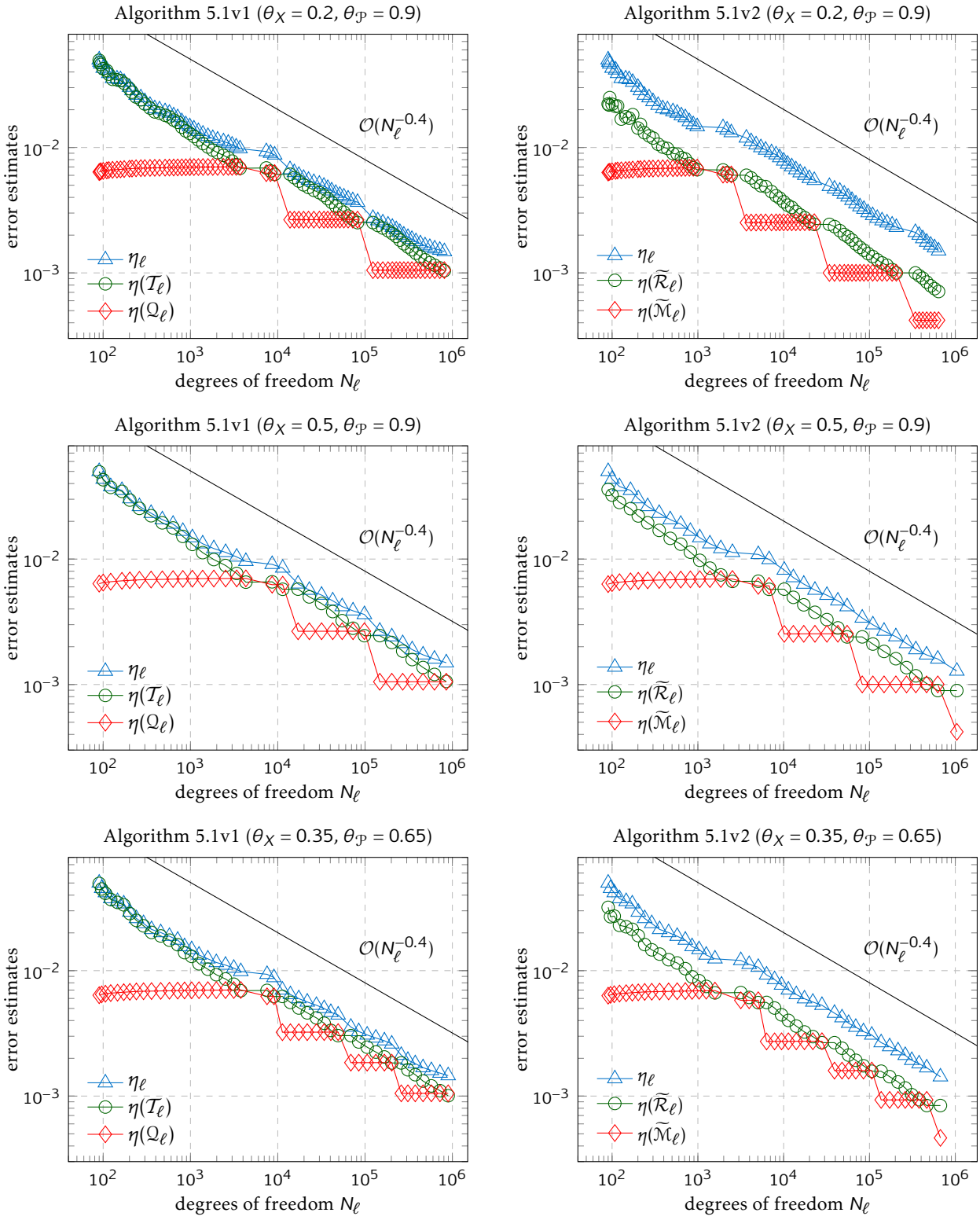


Figure 5.12. Total and local error estimates at each step of Algorithms 5.1v1 and 5.1v2 for the model problem in Section 5.4.3.

Figure 5.12 show the decay of the computed energy error estimates for all computations. For both versions and all sets of parameters, initially, the algorithm performs mesh-refinements to

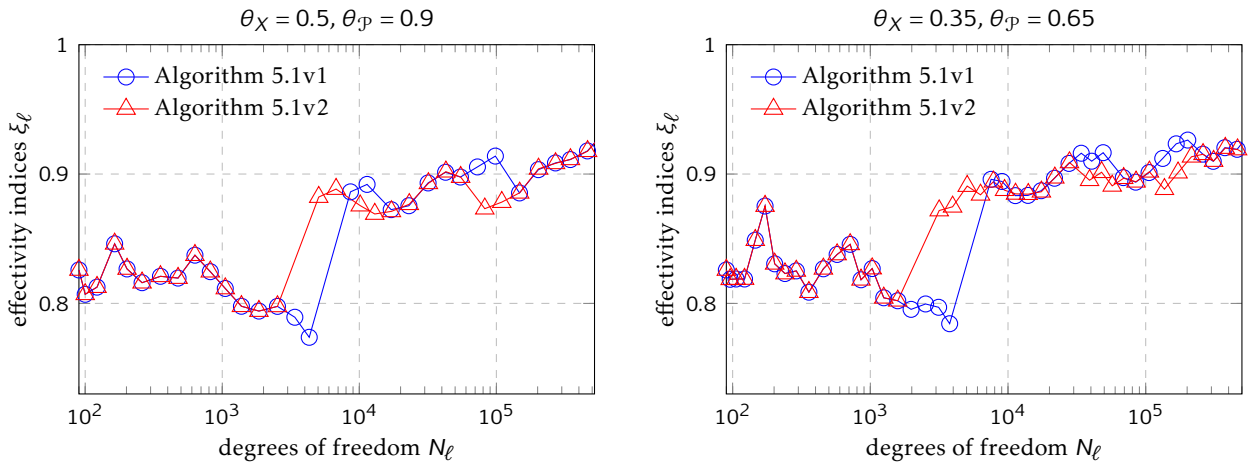


Figure 5.13. The effectivity indices for the SGFEM solutions of the model problem in Section 5.4.4 computed by Algorithms 5.1v1 and 5.1v2 with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$ (case (ii), left) and $\theta_X = 0.35$, $\theta_{\mathcal{P}} = 0.65$ (case (iii), right). The energy norm of the associated reference solution is $\|u_{\text{ref}}\|_B = 1.639745\text{e-}01$.

take account the source of error due to the spatial singularity near the crack tip. Then, a first parametric enrichment, which yields a very small error reduction, in turn enforces a second enrichment after few iterations. Note that the value of $\theta_{\mathcal{P}} = 0.9$, as in cases (i) and (ii), was not eventually large enough to mark more indices, hence yielding larger error reductions. We see that Algorithms 5.1v2 converges faster than Algorithms 5.1v1 during mesh-refinement steps in all three cases; this has been already observed in numerical experiments of previous sections and it is due to the use of Criterion 5.2. On the other hand, the overall decay rate obtained for total error estimates η_ℓ, in all cases, is about $\mathcal{O}(N^{-0.4})$. Notice that this rate is closer to the optimal rate (of $\mathcal{O}(N^{-1/2})$) than the rates achieved by both Algorithms 5.1v1 and 5.1v2 in the numerical experiments of Sections 5.4.2 and 5.4.3.

Finally, focusing on cases (ii) and (iii), we compute the effectivity indices ξ_ℓ defined in (5.43). In this experiment, we employ a reference Galerkin solution u_{ref} computed over the triangulation $\mathcal{T}_{\text{ref}}$ being the uniform refinement of $\mathcal{T}_L$ produced by Algorithm 5.1v1 in case (iii) and using the reference index set $\mathcal{P}_{\text{ref}}$ to be equal to the index set $\mathcal{P}_L$ generated by Algorithm 5.1v2 in case (ii). The resulting effectivity indices are plotted in Figure 5.13. As in experiments of previous Sections 5.4.2 and 5.4.3, the effectivity indices are less than unity for all iterations; for this model problem, however, they increase as iterations progress and tend to be close to 0.9.

The results of this experiment lead us to the same conclusions about the efficiency of Algorithm 5.1v2 as previous experiments in Sections 5.4.2 and 5.4.3 did: provided that a sufficiently small spatial marking parameter θ_X is selected (e.g., $\theta_X = 0.2$ or $\theta = 0.35$), Criterion 5.2 enables

the algorithm to reach the tolerance faster and the total error estimates decay with almost optimal rates during mesh-refinement steps. Interestingly, in this experiment, Algorithm 5.1v2 in case (iii), i.e., with $\theta_{\mathcal{P}} = 0.65$, performed as well as the algorithm in case (i) where $\theta_{\mathcal{P}} = 0.9$. This shows that, effectively, comparison of the versions of the algorithm is problem-dependent and there is no a priori knowledge on the values of input marking parameters that yield 'optimal' results with respect to, for example, computational times and final number of elements.

Chapter 6

Adaptive algorithms driven by two-level a posteriori error estimates

The theoretical analysis of adaptive finite element algorithms has received a remarkable attention for deterministic problems and, recently, also for model problems with parametric or uncertain inputs. Unlike the design of efficient adaptive strategies, however, the convergence analysis of adaptive algorithms for problems with random inputs is less developed. In this chapter, our goal is twofold: introduce a novel a posteriori estimate for the energy error of solutions to parametric PDEs and provide a rigorous convergence analysis of the associated adaptive SGFEM algorithms.

In the first part of this chapter, we introduce an energy error estimate composed of *two-level* a posteriori error estimates for spatial approximations and hierarchical a posteriori error estimates for parametric approximations. The resulting total estimate, that combines these two contributions (see (6.5) below), has been recently introduced and analysed in [26]; the definition and analysis of this new estimate are reported in Section 6.1. Building on the hierarchical framework developed in [24, 29] and recalled in Section 5.1, the construction of this estimate for the energy error estimation of model problem (4.5) is based on ideas from [100, 99, 62, 66] (see also [50, 32] for earlier works in this direction). In particular, our first result is the proof of the efficiency and reliability of the estimate (see Theorem 6.1). One of the key advantages of this a posteriori error estimate is that it avoids computing the solution of linear systems when estimating the errors coming from spatial approximations (while keeping the hierarchical structure of the estimate) and thus speeds up the computation. That is, with reference to hierarchical estimate $\eta_{X\mathcal{P}}$ defined in (5.22), this new estimate does not require the solution of the linear system arising from problem (5.20) which is instead needed for computing the contributing spatial estimator $e_{Y\mathcal{P}} \in V_{Y\mathcal{P}}$.

In the second part of the chapter, we focus on the convergence analysis of adaptive SGFEM

algorithms driven by the proposed new error estimate (see Algorithm 6.1). This is taken from the recent work [25]. The algorithm has four versions which employ four different marking criteria that are combinations of the Dörfler and the maximum marking strategies (see Section 2.3.1). At each step, all versions of the algorithm perform either solely mesh-refinement or solely polynomial enrichment (cf. Algorithm 5.1). The central result in Theorem 6.2 shows that each proposed version of the adaptive algorithm generates a sequence of Galerkin approximations such that the associated sequence of energy error estimates converges to zero. Therefore, this result provides a theoretical guarantee that, for any given positive tolerance, all versions of the algorithm stop after a finite number of iterations. As an immediate consequence of Theorem 6.2, we show that, under saturation assumption (5.14), the Galerkin approximations generated by the algorithms converge to the true parametric solution (see Corollary 6.2). Further to that, in the case of Dörfler marking, we prove linear convergence of the computed energy error estimates in Theorem 6.3.

The results of two numerical experiments are reported in Section 6.4. In the first experiment, we compare the performance of the proposed algorithm with Algorithm 5.1 driven by hierarchical error estimates whereas, in the second experiment, we compare the performance of the proposed algorithm with respect to the computational cost associated with using different marking criteria.

6.1 Two-level error estimate

Let $X := \mathcal{S}^1_0(\mathcal{T})$ be the first-order finite element space associated with a conforming triangulation $\mathcal{T}$ of $D \subset \mathbb{R}^2$ and let $\widehat{X} := \mathcal{S}^1_0(\widehat{\mathcal{T}})$ be the enriched space associated with the uniformly refined triangulation $\widehat{\mathcal{T}}$ obtained by NVB refinements (see Section 2.3.2). Let $Y \subset H^1_0(D)$ be the corresponding first-order detail finite element space satisfying (5.3).

We introduce the following notation. Let $N_Y := \mathsf{dim}(Y)$ denote the dimension of Y. Let

$$\mathcal{N}^+ := \mathcal{N}^\circ(\widehat{\mathcal{T}}) \setminus \mathcal{N}^\circ(\mathcal{T}) = \{z_1, \ldots, z_{N_Y}\} \subset \mathcal{N}^\circ(\widehat{\mathcal{T}}), \tag{6.1}$$

be the set of N_Y interior vertices introduced by the uniform refinement of $\mathcal{T}$, i.e., the set of midpoints of the interior edges of $\mathcal{T}$. We denote by $\mathcal{B}_Y := \{\psi_1, \ldots, \psi_{N_Y}\}$ the basis of Y, i.e., for each midpoint $z_j \in \mathcal{N}^+$, $\psi_j \in \mathcal{B}_Y$ is the corresponding piecewise linear function such that $\psi_j(z_j) = 1$ and $\psi_j(\mathsf{x}) = 0$ for all $\mathsf{x} \in \mathcal{N}(\widehat{\mathcal{T}}) \setminus \{z_j\}$, $j = 1, \ldots, N_Y$ (see Figure 5.1(b)). We also define the following one-dimensional subspaces

$$Y_j := \mathsf{span}\{\psi_j\} \qquad \forall\, j = 1, \ldots, N_Y, \tag{6.2}$$

which yield the following decomposition, $Y = \bigoplus_{j=1}^{N_Y} Y_j$. For later use, we note that there exists a finite constant $K \geq 1$ depending on $\mathcal{T}$ such that (see Remark 6.1.2)

$$\#\Big\{\psi_j \in \mathcal{B}_Y \,:\, \text{interior}\Big(\text{supp}(\psi_j) \cap T\Big) \neq \emptyset\Big\} \,\leq\, K \qquad \forall\, T \in \mathcal{T}. \tag{6.3}$$

Furthermore, due to the tensor product structure of $V_{Y\mathcal{P}}$ (see (5.7)), each function $v \in V_{Y\mathcal{P}}$ can be written as

$$v(\mathbf{x},\mathbf{y}) = \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} v_{j\nu}(\mathbf{x}) P_\nu(\mathbf{y}), \qquad \mathbf{x} \in D,\, \mathbf{y} \in \Gamma, \tag{6.4}$$

where $v_{j\nu} \in Y_j$ and $P_\nu \in \mathcal{P}_\mathcal{P}$, for all $\nu \in \mathcal{P}$.

Hereafter, we will write $\lesssim$ to denote $\leq$ up to some positive constant C and, given two quantities a and b, we will write $a \simeq b$ to abbreviate $b \lesssim a \lesssim b$.

6.1.1 Main result

Let $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ be the Galerkin approximation satisfying (4.36) for a given finite index set $\mathcal{P} \subset \mathcal{I}$. Let $\mathcal{Q} \subset \mathcal{I} \setminus \mathcal{P}$ be the associated finite detail index set; this can be constructed, e.g., as in (5.35).

Consider the following quantity

$$\tau^2_{X\mathcal{P}} \,:=\, \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \frac{|F(\psi_j P_\nu) - B(u_{X\mathcal{P}}, \psi_j P_\nu)|^2}{\| a_0^{1/2} \nabla \psi_j \|^2_{L^2(D)}} \,+\, \sum_{\mu\in\mathcal{Q}} \| e^{(\mu)}_{X\mathcal{Q}} \|^2_{B_0}, \tag{6.5}$$

where $\psi_j \in \mathcal{B}_Y$ and $e^{(\mu)}_{X\mathcal{Q}}$ are the individual estimators satisfying (5.27) for all $\mu \in \mathcal{Q}$. Note that $\psi_j P_\nu \in V_{Y\mathcal{P}}$ for all $\psi_j \in \mathcal{B}_Y$ and all $\nu \in \mathcal{P}$.

Theorem 6.1. *Let $u \in V$ be the solution to problem* (4.17)*, and let $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ and $\widehat{u}_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$ be two Galerkin approximations satisfying* (4.36) *and* (5.10)*, respectively. There exists a constant $C_{thm} \geq 1$, which depends only on the shape regularity of $\mathcal{T}$ and $\widehat{\mathcal{T}}$, the mesh-refinement rule, and the mean field a_0 in* (4.8)*, such that $\tau_{X\mathcal{P}}$ defined in* (6.5) *satisfies*

$$\frac{\lambda}{K} \tau^2_{X\mathcal{P}} \,\leq\, \| \widehat{u}_{X\mathcal{P}} - u_{X\mathcal{P}} \|^2_B \,\leq\, \Lambda\, C_{thm}\, \tau^2_{X\mathcal{P}}, \tag{6.6}$$

where λ and Λ are the constants in (4.22) *and K is the constant in* (6.3)*. Furthermore, under saturation assumption* (5.14) *with constant $q_{\text{sat}} \in [0,1)$, there holds*

$$\frac{\lambda}{K} \tau^2_{X\mathcal{P}} \,\leq\, \| u - u_{X\mathcal{P}} \|^2_B \,\leq\, \frac{\Lambda\, C_{thm}}{1 - q^2_{\text{sat}}}\, \tau^2_{X\mathcal{P}}. \tag{6.7}$$

On the one hand, Theorem 6.1 shows that $\tau_{X\mathcal{P}}$ is an efficient and reliable a posteriori estimate for the energy norm of the error $e = u - u_{X\mathcal{P}}$ (see (6.7)). On the other hand, recall that $\| \widehat{u}_{X\mathcal{P}} - u_{X\mathcal{P}} \|_B$

is the error reduction (in the energy norm) that would be achieved if the enhanced solution $\widehat{u}_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$ were to be computed (see (5.13)). Hence, inequalities in (6.6) show that $\tau_{X\mathcal{P}}$ also provides an estimate for this error reduction. Moreover, we stress that Theorem 6.1 holds, indeed, for any finite detail index set $\mathcal{Q} \subset \mathcal{I} \setminus \mathcal{P}$ and any conforming refinement $\widehat{\mathcal{T}}$ of $\mathcal{T}$ (and corresponding detail space Y); see, e.g., Figure 5.1(a) for the case of regular refinements. In addition, we also emphasise that the proof applies for any spatial dimension, while we restrict ourselves to the two-dimensional case $D \subset \mathbb{R}^2$ for ease of presentation. The proof of Theorem 6.1 is postponed to Section 6.1.3.

The structure of the a posteriori error estimate $\tau_{X\mathcal{P}}$ defined in (6.5) is similar to that of the hierarchical estimate $\eta_{X\mathcal{P}}$. In fact, while the the parametric part of both estimates is the same (cf. (5.22) and (6.5)), they only differ in the spatial contribution. Such contribution is adapted from what in the literature is typically referred to as *two-level* error estimate; see, e.g., [100, 99, 62]. In order to give a proper name to estimate $\tau_{X\mathcal{P}}$ defined in (6.5) and distinguish it from hierarchical estimate $\eta_{X\mathcal{P}}$ defined in (5.22), in the reminder of the thesis, we keep the terminology for the spatial contribution and refer to $\tau_{X\mathcal{P}}$ as the *two-level* a posteriori error estimate.

Remark 6.1.1. *The spatial contributing part of $\tau_{X\mathcal{P}}$ defined in* (6.5) *includes:*

- *in the numerator, the entries of the* residual *of $u_{X\mathcal{P}}$, where the Galerkin data are computed with respect to the enriching space $V_{Y\mathcal{P}}$ (cf. the right-hand side of* (5.20)*);*
- *in the denominator, the diagonal entries of the spatial finite element matrix associated with the detail space Y, since*

$$\| a_0^{1/2} \nabla \psi_j \|^2_{L^2(D)} = \int_D a_0(\mathbf{x}) \nabla \psi_j(\mathbf{x}) \cdot \nabla \psi_j(\mathbf{x}) d\mathbf{x} \qquad \forall\, j = 1, \dots, N_Y.$$

  Moreover, the denominator can be easily simplified. For every $T \in \mathcal{T}$ with diameter h_T (see (2.8)*), let $\psi_j \in \mathcal{B}_Y$ such that $\mathrm{supp}(\psi_j) \subseteq \omega(T)$ (see* (2.7)*). Then, there holds*

$$\| a_0^{1/2} \nabla \psi_j \|^2_{L^2(D)} \simeq h_T^{-2} \| \psi_j \|^2_{L^2(D)} \simeq 1,$$

  where hidden constants depend only on the shape regularity of $\widehat{\mathcal{T}}$, the (local) mesh-refinement rule, and the mean field a_0 in (4.8)*. For instance, if $a_0 := 1$, elementary calculations show that $\| a_0^{1/2} \nabla \psi_j \|^2_{L^2(D)} = 4$ in case $\widehat{\mathcal{T}}$ is obtained by uniform NVB refinements.*

Thus, the implementation of the spatial estimate contributing to $\tau_{X\mathcal{P}}$ only requires the assembly of the residual of the Galerkin solution $u_{X\mathcal{P}}$ on $V_{Y\mathcal{P}}$ and no linear system needs to be solved.

Remark 6.1.2. *Note that in the two-dimensional case, the number $N_\gamma = \#\mathcal{N}^+$ of interior midpoints introduced by the uniform refinement of $\mathcal{T}$ is equal to the number $\#\mathcal{E}^\circ(\mathcal{T})$ of interior edges of $\mathcal{T}$, i.e., trivially, there exists a one-to-one map between $\mathcal{N}^+$ and $\mathcal{E}^\circ(\mathcal{T})$. In particular, it is easy to see that $K = 3$ in* (6.3).

6.1.2 Auxiliary lemmas

In this section, we collect four auxiliary results which are required to prove Theorem 6.1. Let us start with the following observation.

Lemma 6.1. *For all $v, w \in V$, the following equality holds*

$$B_0(v,w) = \sum_{\nu\in\mathcal{J}} \int_D a_0(\mathbf{x})\nabla v_\nu(\mathbf{x})\cdot\nabla w_\nu(\mathbf{x})d\mathbf{x} \qquad \forall\ v_\nu, w_\nu \in H_0^1(D). \tag{6.8}$$

In particular, it follows that

$$\|v\|_{B_0}^2 = \sum_{\nu\in\mathcal{J}} \|a_0^{1/2}\nabla v_\nu\|_{L^2(D)}^2. \tag{6.9}$$

Proof. Using the polynomial chaos expansion (4.30) for both v and w in V, we have

$$\begin{aligned} B_0(v,w) &= \sum_{\nu\in\mathcal{J}}\sum_{\mu\in\mathcal{J}} \int_\Gamma\int_D a_0(\mathbf{x})\nabla v_\nu(\mathbf{x})\cdot\nabla w_\mu(\mathbf{x})\,P_\nu(\mathbf{y})P_\mu(\mathbf{y})\,d\mathbf{x}\,d\pi(\mathbf{y}) \\ &= \sum_{\nu\in\mathcal{J}}\sum_{\mu\in\mathcal{J}} \int_D a_0(\mathbf{x})\nabla v_\nu(\mathbf{x})\cdot\nabla w_\mu(\mathbf{x})\,d\mathbf{x} \underbrace{\int_\Gamma P_\nu(\mathbf{y})P_\mu(\mathbf{y})\,d\pi(\mathbf{y})}_{=(P_\nu,P_\mu)_\pi}. \end{aligned}$$

Since polynomials $\{P_\nu\}_{\nu\in\mathcal{J}}$ are orthonormal in $L^2_\pi(\Gamma)$ with respect to inner product $(\cdot,\cdot)_\pi$, this proves (6.8). Moreover, (6.9) follows by choosing $w = v$ in (6.8). □

Lemma 6.2. *Consider a function $v \in V_{\gamma\mathcal{P}}$ and its representation* (6.4)*. There holds:*

$$K^{-1}\|v\|_{B_0}^2 \;\le\; \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_\gamma} \|a_0^{1/2}\nabla v_{j\nu}\|_{L^2(D)}^2 \;\le\; C_{loc}\|v\|_{B_0}^2, \tag{6.10}$$

where the constant $C_{loc} > 0$ depends only on the shape regularity of $\widehat{\mathcal{T}}$, the (local) mesh-refinement rule, and the mean field a_0 in (4.8).

Proof. We divide the proof into three steps.

Step 1. Let $T \in \mathcal{T}$ and consider a function $w_j \in Y_j$, where Y_j is defined in (6.2) for all $j =$

$1,\dots,N_Y$. Observe that

$$\left\| a_0^{1/2}\nabla\left(\sum_{j=1}^{N_Y} w_j\right)\right\|_{L^2(T)} \le \sum_{j=1}^{N_Y} \| a_0^{1/2}\nabla w_j\|_{L^2(T)} \le \sqrt{K}\left(\sum_{j=1}^{N_Y} \| a_0^{1/2}\nabla w_j\|^2_{L^2(T)}\right)^{1/2}.$$

Hence, summing over all elements $T\in\mathcal{T}$, we obtain

$$\left\| a_0^{1/2}\nabla\left(\sum_{j=1}^{N_Y} w_j\right)\right\|^2_{L^2(D)} \le K\sum_{T\in\mathcal{T}}\sum_{j=1}^{N_Y} \| a_0^{1/2}\nabla w_j\|^2_{L^2(T)} = K\sum_{j=1}^{N_Y} \| a_0^{1/2}\nabla w_j\|^2_{L^2(D)}. \tag{6.11}$$

Step 2. We now prove the converse estimate of (6.11). For a function $w_Y\in Y$, which can be represented as $w_Y=\sum_{j=1}^{N_Y} w_j$ with unique coefficients $w_j\in Y_j$, for all $j=1,\dots,N_Y$, notice that the following two quantities

$$\| w_Y|_T\|_{Y,1} := \left(\sum_{j=1}^{N_Y} \| a_0^{1/2}\nabla w_j\|^2_{L^2(T)}\right)^{1/2} \quad\text{and}\quad \| w_Y|_T\|_{Y,2} := \left\| a_0^{1/2}\nabla\left(\sum_{j=1}^{N_Y} w_j\right)\right\|_{L^2(T)},$$

where $T\in\mathcal{T}$, define two norms on the subspace $Y|_T := \{w_Y|_T \,:\, w_Y\in Y\}$. For example, one trivially has that $\|\lambda w_Y|_T\|_{Y,1} = \lambda\| w_Y|_T\|_{Y,1}$ for all $\lambda\in\mathbb{R}$. For some $z_Y=\sum_{j=1}^{N_Y} z_j\in Y$, there holds

$$\begin{aligned}\| w_Y|_T + z_Y|_T\|_{Y,1} &\le \left(\sum_{j=1}^{N_Y} \| a_0^{1/2}\nabla(w_j+z_j)\|^2_{L^2(T)}\right)^{1/2}\\ &\le \left(\sum_{j=1}^{N_Y} \| a_0^{1/2}\nabla w_j\|^2_{L^2(T)} + \sum_{j=1}^{N_Y} \| a_0^{1/2}\nabla z_j\|^2_{L^2(T)}\right)^{1/2} \le \| w_Y|_T\|_{Y,1} + \| z_Y|_T\|_{Y,1}.\end{aligned}$$

Furthermore, note that $\| a_0^{1/2}\nabla w_j\|_{L^2(T)}=0$ for all $w_j\in Y_j$ implies $w_Y|_T=0$ and so $\| w_Y|_T\|_{Y,1}$. Analogous arguments hold for norm $\|\cdot\|_{Y,2}$. Due to equivalence of norms on finite-dimensional spaces (see, e.g., [92, p. 32]), we then conclude that

$$\sum_{j=1}^{N_Y} \left\| a_0^{1/2}\nabla w_j\right\|^2_{L^2(T)} \simeq \left\| a_0^{1/2}\nabla\left(\sum_{j=1}^{N_Y} w_j\right)\right\|^2_{L^2(T)} \quad \forall\, w_j\in Y_j,\ j=1,\dots,N_Y, \tag{6.12}$$

where the equivalence constants depend on the mean field a_0 in (4.8), the shape regularity of $\widehat{\mathcal{T}}$, as well as on the type of the mesh-refinement rule (that affects the configuration of the local space $Y|_T$). Summing over all elements $T\in\mathcal{T}$ in (6.12), we obtain, in particular, the following upper bound

$$\sum_{j=1}^{N_Y} \| a_0^{1/2}\nabla w_j\|^2_{L^2(D)} \le C_{\mathrm{loc}}\left\| a_0^{1/2}\nabla\left(\sum_{j=1}^{N_Y} w_j\right)\right\|^2_{L^2(D)} \quad \forall\, w_j\in Y_j,\ j=1,\dots,N_Y. \tag{6.13}$$

Step 3. Let $v\in V_{Y\mathcal{P}}$ represented as in (6.4). From Lemma 6.1, it holds that

$$\| v\|^2_{B_0} \overset{(6.9)}{=} \sum_{\nu\in\mathcal{P}} \left\| a_0^{1/2}\nabla\left(\sum_{j=1}^{N_Y} v_{j\nu}\right)\right\|^2_{L^2(D)},$$

with $v_{j\nu} \in Y_j$ for all $\nu \in \mathcal{P}$ and $j = 1,\dots,N_Y$. Then, the two-sides bound (6.10) follows using the foregoing estimates (6.11) and (6.13) from Step 1 and Step 2, respectively. □

To state the next lemma, we recall the nodal interpolation operator $\mathcal{I}_{\mathcal{T}} : C(\overline{D}) \to X$ defined by

$$\mathcal{I}_{\mathcal{T}} v(\mathbf{x}) := \sum_{i=1}^{N_X} v(\mathbf{x}_i)\varphi_i(\mathbf{x}), \qquad \mathbf{x} \in D, \tag{6.14}$$

where $\mathbf{x}_i \in \mathcal{N}^\circ(\mathcal{T})$ and $\varphi_i \in X$ is the hat function associated with $\mathbf{x}_i$, $i = 1,\dots,N_X$. Note that $\mathcal{I}_{\mathcal{T}} v$ is the unique function in $\mathcal{S}^1_0(\mathcal{T})$ that has the same nodal values as v (see, e.g., [35, Section 3.3]). Now, if $V_{\widehat{X}\mathcal{P}} := \widehat{X} \otimes \mathcal{P}_{\mathcal{P}}$ denotes the enriched space spanned by functions of the form

$$v_{\widehat{X}\mathcal{P}}(\mathbf{x},\mathbf{y}) = \sum_{\nu\in\mathcal{P}} \widehat{v}_\nu(\mathbf{x})P_\nu(\mathbf{y}) \quad \text{with unique coefficients } \widehat{v}_\nu \in \widehat{X}, \tag{6.15}$$

then we have that $\mathcal{I}_{\mathcal{T}}\widehat{v}_\nu \in X$ and we can represent a function $v \in V_{X\mathcal{P}}$ as

$$v_{X\mathcal{P}}(\mathbf{x},\mathbf{y}) = \sum_{\nu\in\mathcal{P}} (\mathcal{I}_{\mathcal{T}}\widehat{v}_\nu(\mathbf{x}))\, P_\nu(\mathbf{y}), \qquad \mathbf{x} \in D,\ \mathbf{y} \in \Gamma. \tag{6.16}$$

Lemma 6.3. *Let $\mathcal{I}_{\mathcal{T}}$ be the nodal interpolation operator* (6.14). *Let $v_{\widehat{X}\mathcal{P}} \in V_{\widehat{X}\mathcal{P}}$ and $v_{X\mathcal{P}} \in V_{X\mathcal{P}}$ be given by* (6.15) *and* (6.16), *respectively. Then, we have the representation*

$$v_{\widehat{X}\mathcal{P}} - v_{X\mathcal{P}} = \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y} v_{j\nu}P_\nu \in V_{Y\mathcal{P}} \qquad \text{with } v_{j\nu} \in Y_j \text{ for all } \nu \in \mathcal{P}, \tag{6.17}$$

and there holds

$$\| v_{\widehat{X}\mathcal{P}} - v_{X\mathcal{P}} \|_{B_0} \le C_{\mathrm{stb}} \| v_{\widehat{X}\mathcal{P}} \|_{B_0}, \tag{6.18}$$

where the constant $C_{\mathrm{stb}} > 0$ depends only on the shape regularity of $\widehat{\mathcal{T}}$, the (local) mesh-refinement rule, and the mean field a_0 in (4.8).

Proof. The proof consists of two steps.

Step 1. Consider a function $v_X := \mathcal{I}_{\mathcal{T}} v_{\widehat{X}} \in X$ where $v_{\widehat{X}} \in \widehat{X}$. Since $\widehat{X} = X \oplus Y$, there exist unique $w_X \in X$ and $w_Y \in Y$ such that the function $v_{\widehat{X}} - v_X \in \widehat{X}$ can be represented as $v_{\widehat{X}} - v_X = w_X + w_Y$. Since $v_{\widehat{X}}$ and v_X coincide on every vertex $\mathbf{x}_{\mathcal{T}} \in \mathcal{N}(\mathcal{T})$ of triangulation $\mathcal{T}$, and functions of Y vanish on $\mathbf{x}_{\mathcal{T}}$, i.e., we have that $0 = (v_{\widehat{X}} - v_X)(\mathbf{x}_{\mathcal{T}}) = w_X(\mathbf{x}_{\mathcal{T}})$ for all $\mathbf{x}_{\mathcal{T}} \in \mathcal{N}(\mathcal{T})$. Thus, $w_X = 0$, and this implies that $v_{\widehat{X}} - v_X \in Y$. Moreover, using standard scaling arguments (see, e.g., [35]), we have that

$$\| a_0^{1/2}\nabla(\mathcal{I}_{\mathcal{T}} v_{\widehat{X}})\|_{L^2(T)} \lesssim \| a_0^{1/2}\nabla v_{\widehat{X}}\|_{L^2(T)} \qquad \forall\, T \in \mathcal{T},\ \forall\, v_{\widehat{X}} \in \widehat{X},$$

where hidden constant depend only on the mean field a_0 in (4.8) and the shape regularity of $\widehat{\mathcal{T}}$, as well as on the type of the mesh-refinement strategy (that affects the configuration of the local

space $Y|_T$). Summing this estimate over all $T \in \mathcal{T}$, we obtain

$$\|a_0^{1/2}\nabla(\mathcal{I}_{\mathcal{T}}\, v_{\widehat{X}})\|_{L^2(D)} \lesssim \|a_0^{1/2}\nabla v_{\widehat{X}}\|_{L^2(D)} \qquad \forall\, v_{\widehat{X}} \in \widehat{X}. \tag{6.19}$$

Step 2. Recall that $v_{\widehat{X}\mathcal{P}} - v_{X\mathcal{P}} = \sum_{\nu\in\mathcal{P}}(\widehat{v}_\nu - \mathcal{I}_{\mathcal{T}}\,\widehat{v}_\nu)P_\nu$ with $\widehat{v}_\nu \in \widehat{X}$ and $\mathcal{I}_{\mathcal{T}}\,\widehat{v}_\nu \in X$ for all $\nu \in \mathcal{P}$ (cf. (6.15) and (6.16)). According to Step 1, we have that $\widehat{v}_\nu - \mathcal{I}_{\mathcal{T}}\,\widehat{v}_\nu \in Y$ and hence $\widehat{v}_\nu - \mathcal{I}_{\mathcal{T}}\,\widehat{v}_\nu = \sum_{j=1}^{N_Y} v_{j\nu} \in Y$, with some $v_{j\nu} \in Y_j$, for all $\nu \in \mathcal{P}$. This proves (6.17). Moreover, representations (6.15) and (6.16) yield

$$\|v_{X\mathcal{P}}\|_{B_0}^2 \overset{(6.9)}{=} \sum_{\nu\in\mathcal{P}} \|a_0^{1/2}\nabla(\mathcal{I}_{\mathcal{T}}\,\widehat{v}_\nu)\|_{L^2(D)}^2 \overset{(6.19)}{\lesssim} \sum_{\nu\in\mathcal{P}} \|a_0^{1/2}\nabla\widehat{v}_\nu\|_{L^2(D)}^2 \overset{(6.9)}{=} \|v_{\widehat{X}\mathcal{P}}\|_{B_0}^2.$$

The triangle inequality then proves (6.18). □

For the last lemma, we introduce some further notation. Let $\mathcal{G}_{X\mathcal{P}} : V \to V_{X\mathcal{P}}$ be the orthogonal projection onto $V_{X\mathcal{P}}$ with respect to $B_0(\cdot,\cdot)$, i.e., for all $w \in V$, $\mathcal{G}_{X\mathcal{P}}$ satisfies

$$B_0(w, v_{X\mathcal{P}}) = B_0(\mathcal{G}_{X\mathcal{P}}w, v_{X\mathcal{P}}) \qquad \forall\, v_{X\mathcal{P}} \in V_{X\mathcal{P}}. \tag{6.20}$$

Furthermore, for all $\nu \in \mathcal{P}$ and all $j = 1,\dots,N_Y$, let $\mathcal{G}_{j\nu} : V \to V_{j\nu}$ be the orthogonal projection onto the subspace $V_{j\nu} := \operatorname{span}\{\psi_j P_\nu\}$ for $\psi_j \in B_Y$ and $P_\nu \in \mathcal{P}_\nu$, with respect to $B_0(\cdot,\cdot)$, i.e., for all $w \in V$, $\mathcal{G}_{j\nu}$ satisfies

$$B_0(w, vP_\nu) = B_0(\mathcal{G}_{j\nu}w, vP_\nu) \qquad \forall\, vP_\nu \in V_{j\nu}. \tag{6.21}$$

Lemma 6.4. *For any $v_{\widehat{X}\mathcal{P}} \in V_{\widehat{X}\mathcal{P}}$, the following estimates hold*

$$C_Y^{-1}\|v_{\widehat{X}\mathcal{P}}\|_{B_0}^2 \le \|\mathcal{G}_{X\mathcal{P}}v_{\widehat{X}\mathcal{P}}\|_{B_0}^2 + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\|\mathcal{G}_{j\nu}v_{\widehat{X}\mathcal{P}}\|_{B_0}^2 \le 2K\|v_{\widehat{X}\mathcal{P}}\|_{B_0}^2, \tag{6.22}$$

where the constant $C_Y \ge 1$ depends only on the shape regularity of $\widehat{\mathcal{T}}$, the (local) mesh-refinement rule, and the mean field a_0 in (4.8)*. Moreover, if $\mathcal{G}_{X\mathcal{P}}v_{\widehat{X}\mathcal{P}} = 0$, the upper bound in* (6.22) *holds with constant K (instead of $2K$).*

Proof. We divide the proof into two steps.

Step 1. Let us first prove the lower bound in (6.22). To this end, let $v_{\widehat{X}\mathcal{P}} \in V_{\widehat{X}\mathcal{P}}$ and $v_{X\mathcal{P}} \in V_{X\mathcal{P}}$.

From Lemma 6.3, we have that

$$\begin{aligned}\|v_{\widehat{X}\mathcal{P}}\|_{B_0}^2 &\overset{(6.17)}{=} B_0(v_{\widehat{X}\mathcal{P}}, v_{X\mathcal{P}}) + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y} B_0(v_{\widehat{X}\mathcal{P}}, v_{j\nu}P_\nu)\\ &= B_0(\mathcal{G}_{X\mathcal{P}}v_{\widehat{X}\mathcal{P}}, v_{X\mathcal{P}}) + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y} B_0(\mathcal{G}_{j\nu}v_{\widehat{X}\mathcal{P}}, v_{j\nu}P_\nu)\\ &\leq \left(\|\mathcal{G}_{X\mathcal{P}}v_{\widehat{X}\mathcal{P}}\|_{B_0}^2 + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\|\mathcal{G}_{j\nu}v_{\widehat{X}\mathcal{P}}\|_{B_0}^2\right)^{1/2}\left(\|v_{X\mathcal{P}}\|_{B_0}^2 + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\|v_{j\nu}P_\nu\|_{B_0}^2\right)^{1/2},\end{aligned}$$

where the second equality follows by (6.20) and (6.21). We now find an upper bound for the second term in brackets in the right-hand side of the inequality above. First, note that

$$\|v_{X\mathcal{P}}\|_{B_0} \leq \|v_{\widehat{X}\mathcal{P}}\|_{B_0} + \|v_{\widehat{X}\mathcal{P}} - v_{X\mathcal{P}}\|_{B_0} \overset{(6.18)}{\leq} (1 + C_{\mathrm{stb}})\|v_{\widehat{X}\mathcal{P}}\|_{B_0}.$$

Second, the upper bound in (6.10) from Lemma 6.2 leads us to

$$\begin{aligned}\sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\|v_{j\nu}P_\nu\|_{B_0}^2 &\overset{(6.9)}{=} \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\|a_0^{1/2}\nabla v_{j\nu}\|_{L^2(D)}^2\\ &\overset{(6.10)}{\leq} C_{\mathrm{loc}}\Big\|\sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y} v_{j\nu}P_\nu\Big\|_{B_0}^2\\ &\overset{(6.17)}{=} C_{\mathrm{loc}}\|v_{\widehat{X}\mathcal{P}} - v_{X\mathcal{P}}\|_{B_0}^2 \overset{(6.18)}{\leq} C_{\mathrm{loc}}C_{\mathrm{stb}}^2\|v_{\widehat{X}\mathcal{P}}\|_{B_0}^2.\end{aligned}$$

Combining the foregoing three estimates, we conclude that

$$\|v_{\widehat{X}\mathcal{P}}\|_{B_0}^2 \leq C_Y\left(\|\mathcal{G}_{X\mathcal{P}}v_{\widehat{X}\mathcal{P}}\|_{B_0}^2 + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\|\mathcal{G}_{j\nu}v_{\widehat{X}\mathcal{P}}\|_{B_0}^2\right),$$

where

$$C_Y := (1 + C_{\mathrm{stb}})^2 + C_{\mathrm{loc}}C_{\mathrm{stb}}^2 \geq 1. \tag{6.23}$$

Step 2. Let us now prove the upper bound in (6.22). One has

$$\begin{aligned}\|\mathcal{G}_{X\mathcal{P}}v_{\widehat{X}\mathcal{P}}\|_{B_0}^2 + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\|\mathcal{G}_{j\nu}v_{\widehat{X}\mathcal{P}}\|_{B_0}^2 &= B_0(\mathcal{G}_{X\mathcal{P}}v_{\widehat{X}\mathcal{P}}, v_{\widehat{X}\mathcal{P}}) + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y} B_0(\mathcal{G}_{j\nu}v_{\widehat{X}\mathcal{P}}, v_{\widehat{X}\mathcal{P}})\\ &= B_0\Big(\mathcal{G}_{X\mathcal{P}}v_{\widehat{X}\mathcal{P}} + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\mathcal{G}_{j\nu}v_{\widehat{X}\mathcal{P}}, v_{\widehat{X}\mathcal{P}}\Big) \qquad (6.24)\\ &\leq \Big\|\mathcal{G}_{X\mathcal{P}}v_{\widehat{X}\mathcal{P}} + \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\mathcal{G}_{j\nu}v_{\widehat{X}\mathcal{P}}\Big\|_{B_0}\|v_{\widehat{X}\mathcal{P}}\|_{B_0},\end{aligned}$$

where the first equality holds due to (6.20) and (6.21). Now, firstly observe that

$$\Big\| \mathcal{G}_{X\mathcal{P}} v_{\widehat{X}\mathcal{P}} + \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \mathcal{G}_{j\nu} v_{\widehat{X}\mathcal{P}} \Big\|_{B_0} \leq \sqrt{2} \Big(\| \mathcal{G}_{X\mathcal{P}} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2 + \Big\| \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \mathcal{G}_{j\nu} v_{\widehat{X}\mathcal{P}} \Big\|_{B_0}^2 \Big)^{1/2}. \tag{6.25}$$

The above inequality easily follows by considering the square of its left-hand side, and then using the Cauchy-Schwarz inequality and the fact that $2ab \leq a^2 + b^2$ for all $a, b \in \mathbb{R}$. Next, notice that $\mathcal{G}_{j\nu} v_{\widehat{X}\mathcal{P}} = v_{j\nu} P_\nu \in V_{j\nu}$ for some $v_{j\nu} \in Y_j$ and $\nu \in \mathcal{P}$ (cf. (6.21)). Then,

$$\begin{aligned} \Big\| \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \mathcal{G}_{j\nu} v_{\widehat{X}\mathcal{P}} \Big\|_{B_0}^2 &= \Big\| \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} v_{j\nu} P_\nu \Big\|_{B_0}^2 \\ &\overset{(6.10)}{\leq} K \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \| a_0^{1/2} \nabla v_{j\nu} \|_{L^2(D)}^2 \overset{(6.9)}{=} K \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \| \mathcal{G}_{j\nu} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2. \end{aligned} \tag{6.26}$$

Combining the previous three inequalities we have that

$$\begin{aligned} \| \mathcal{G}_{X\mathcal{P}} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2 + \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \| \mathcal{G}_{j\nu} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2 &\leq \sqrt{2} \Big(\| \mathcal{G}_{X\mathcal{P}} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2 + K \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \| \mathcal{G}_{j\nu} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2 \Big)^{1/2} \| v_{\widehat{X}\mathcal{P}} \|_{B_0} \\ &\leq \sqrt{2K} \Big(\| \mathcal{G}_{X\mathcal{P}} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2 + \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \| \mathcal{G}_{j\nu} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2 \Big)^{1/2} \| v_{\widehat{X}\mathcal{P}} \|_{B_0}, \end{aligned}$$

which yields the upper bound in (6.22):

$$\Big(\| \mathcal{G}_{X\mathcal{P}} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2 + \sum_{\nu\in\mathcal{P}} \sum_{j=1}^{N_Y} \| \mathcal{G}_{j\nu} v_{\widehat{X}\mathcal{P}} \|_{B_0}^2 \Big)^{1/2} \leq \sqrt{2K} \| v_{\widehat{X}\mathcal{P}} \|_{B_0}. \qquad \square$$

This concludes the proof.

6.1.3 Proof of the efficiency and reliability of the two-level estimate

Let $e_{\widehat{X}\mathcal{P}} \in V_{\widehat{X}\mathcal{P}}$ be the unique solution to the residual problem

$$B_0(e_{\widehat{X}\mathcal{P}}, v) = F(v) - B(u_{X\mathcal{P}}, v) \qquad \forall\, v \in V_{\widehat{X}\mathcal{P}}. \tag{6.27}$$

It is easy to see that $e_{\widehat{X}\mathcal{P}}$, defined by (6.27), satisfies

$$B_0(e_{\widehat{X}\mathcal{P}}, v) = B(u_{\widehat{X}\mathcal{P}} - u_{X\mathcal{P}}, v) \qquad \forall\, v \in V_{\widehat{X}\mathcal{P}},$$

where $u_{\widehat{X}\mathcal{P}} \in V_{\widehat{X}\mathcal{P}}$ is the unique Galerkin solution to the corresponding discrete formulation posed on $V_{\widehat{X}\mathcal{P}}$ (see (5.29)). Note that since $\mathcal{P} \cap \mathcal{Q} = \emptyset$, the spaces $V_{\widehat{X}\mathcal{P}}$ and $V_{X\mathcal{Q}}$ are orthogonal with respect to $B_0(\cdot,\cdot)$ (recall that $V_{Y\mathcal{P}} \subset V_{\widehat{X}\mathcal{P}}$ and $V_{X\mathcal{Q}}$ are orthogonal with respect to $B_0(\cdot,\cdot)$, see (5.9)). Therefore, from the observation above and (5.25), we can decompose the estimator $\widehat{e}_{X\mathcal{P}} \in \widehat{V}_{X\mathcal{P}}$ defined by

(5.17) as

$$\widehat{e}_{X\mathcal{P}} \;=\; e_{\widehat{X}\mathcal{P}} \;+\; \sum_{\mu\in\mathcal{Q}} e^{(\mu)}_{X\mathcal{Q}} \qquad \text{with} \qquad \|\widehat{e}_{X\mathcal{P}}\|^2_{B_0} \;=\; \|e_{\widehat{X}\mathcal{P}}\|^2_{B_0} \;+\; \sum_{\mu\in\mathcal{Q}} \|e^{(\mu)}_{X\mathcal{Q}}\|^2_{B_0}\,, \tag{6.28}$$

where $e^{(\mu)}_{X\mathcal{Q}}$ are the individual estimators satisfying (5.27) for all $\mu\in\mathcal{Q}$.

We are now ready to prove Theorem 6.1.

Proof of Theorem 6.1. We divide the proof into two steps.

Step 1. Let $e_{\widehat{X}\mathcal{P}}\in V_{\widehat{X}\mathcal{P}}$ be the unique solution to residual problem (6.27). Since $V_{X\mathcal{P}}\subset V_{\widehat{X}\mathcal{P}}$, we deduce from (4.36) and (6.27) that

$$B_0(e_{\widehat{X}\mathcal{P}}, v) \;=\; 0 \qquad \forall\; v\in V_{X\mathcal{P}}\,.$$

Hence, $\mathcal{G}_{X\mathcal{P}}e_{\widehat{X}\mathcal{P}}=0$ (cf. (6.20)) and therefore, Lemma 6.4, proves that

$$C_Y^{-1}\|e_{\widehat{X}\mathcal{P}}\|^2_{B_0} \;\le\; \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\|\mathcal{G}_{j\nu}e_{\widehat{X}\mathcal{P}}\|^2_{B_0} \;\le\; K\|e_{\widehat{X}\mathcal{P}}\|^2_{B_0}\,.$$

Since $C_Y, K\ge 1$, we use decomposition (6.28) to obtain the following estimates

$$C_Y^{-1}\|\widehat{e}_{X\mathcal{P}}\|^2_{B_0} \;\le\; \sum_{\nu\in\mathcal{P}}\sum_{j=1}^{N_Y}\|\mathcal{G}_{j\nu}e_{\widehat{X}\mathcal{P}}\|^2_{B_0} \;+\; \sum_{\mu\in\mathcal{Q}}\|e^{(\mu)}_{X\mathcal{Q}}\|^2_{B_0} \;\le\; K\|\widehat{e}_{X\mathcal{P}}\|^2_{B_0}\,. \tag{6.29}$$

Step 2. The orthogonal projection $\mathcal{G}_{j\nu}$ onto the one-dimensional space $V_{j\nu}$ (see (6.21)), satisfies

$$\mathcal{G}_{j\nu}v \;=\; \frac{B_0(v,\psi_jP_\nu)}{\|\psi_jP_\nu\|^2_{B_0}}\psi_jP_\nu \qquad \forall\; v\in V,\ \forall\;\nu\in\mathcal{P}.$$

Hence, we have that

$$\begin{aligned}\|\mathcal{G}_{j\nu}e_{\widehat{X}\mathcal{P}}\|^2_{B_0} \;=\; \frac{|B_0(e_{\widehat{X}\mathcal{P}},\psi_jP_\nu)|^2}{\|\psi_jP_\nu\|^2_{B_0}} &\overset{(6.27)}{=} \frac{|F(\psi_jP_\nu)-B(u_{X\mathcal{P}},\psi_jP_\nu)|^2}{\|\psi_jP_\nu\|^2_{B_0}}\\ &\overset{(6.9)}{=} \frac{|F(\psi_jP_\nu)-B(u_{X\mathcal{P}},\psi_jP_\nu)|^2}{\|a_0^{1/2}\nabla\psi_j\|^2_{L^2(D)}}.\end{aligned} \tag{6.30}$$

Using the definition of $\tau_{X\mathcal{P}}$ given in (6.5), estimates (6.29) thus implies that

$$\frac{\lambda}{K}\tau^2_{X\mathcal{P}} \overset{(6.29)}{\le} \lambda\|\widehat{e}_{X\mathcal{P}}\|^2_{B_0} \overset{(5.18)}{\le} \|\widehat{u}_{X\mathcal{P}}-u_{X\mathcal{P}}\|^2_B \overset{(5.18)}{\le} \Lambda\|\widehat{e}_{X\mathcal{P}}\|^2_{B_0} \overset{(6.29)}{\le} \Lambda\, C_Y\,\tau^2_{X\mathcal{P}}.$$

This proves the two-sides bound (6.6) for the error reduction with $C_{\text{thm}}=C_Y$, where C_Y is defined in (6.23). Furthermore, we also have that

$$\frac{\lambda}{K}\tau^2_{X\mathcal{P}} \;\le\; \|\widehat{u}_{X\mathcal{P}}-u_{X\mathcal{P}}\|^2_B \overset{(5.13)}{\le} \|u-u_{X\mathcal{P}}\|^2_B,$$

which is the lower bound in (6.7). On the other hand, the upper bound in (6.7) immediately

follows by saturation assumption (5.14). □

6.2 Adaptive SGFEM algorithms

In this section, we describe an adaptive algorithm driven by the two-level error estimate $\tau_{X\mathcal{P}}$ defined in (6.5) for the energy error estimation of parametric model problem (4.5). Similarly to Algorithm 5.1, the algorithm presented here is based on iterations of standard finite element loop (2.13). Before describing the algorithm, we introduce the general notation that is used in the rest of the chapter.

6.2.1 Local error contributions

For any edge $E \in \mathcal{E}^\circ(\mathcal{T})$, there is a unique $j \in \{1,\dots,N_Y\}$ such that $\boldsymbol{z}_j \in \mathcal{N}^+$ is the midpoint of E. In turn, for each midpoint $\boldsymbol{z}_j \in \mathcal{N}^+$, there is an associated unique basis function $\psi_j \in B_Y$ (see Remark 6.1.2). In what follows, we change the notation so that the spatial contribution of the two-level estimate is indexed by midpoints $\boldsymbol{z} \in \mathcal{N}^+$ rather than by indices $j \in \{1,\dots,N_Y\}$. That is, we define the spatial error estimate contributing to two-level estimate (6.5) as

$$\tau_{Y\mathcal{P}}(\mathcal{N}^+)^2 := \sum_{\boldsymbol{z}\in\mathcal{N}^+} \tau_{Y\mathcal{P}}(\boldsymbol{z})^2 \qquad \text{with} \qquad \tau_{Y\mathcal{P}}(\boldsymbol{z})^2 := \sum_{\nu\in\mathcal{P}} \frac{|F(\psi_{\boldsymbol{z}}P_\nu) - B(u_{X\mathcal{P}}, \psi_{\boldsymbol{z}}P_\nu)|^2}{\| a_0^{1/2}\nabla\psi_{\boldsymbol{z}}\|^2_{L^2(D)}}, \tag{6.31}$$

i.e., $\tau_{Y\mathcal{P}}(\boldsymbol{z})$ denotes the local spatial error estimate associated with the midpoint $\boldsymbol{z} \in \mathcal{N}^+$. Likewise, we define the parametric error estimate contributing to two-level estimate (6.5) as

$$\tau_{X\mathcal{Q}}(\mathcal{Q})^2 := \sum_{\mu\in\mathcal{Q}} \tau_{X\mathcal{Q}}(\mu)^2 \qquad \text{with} \qquad \tau_{X\mathcal{Q}}(\mu) := \| e^{(\mu)}_{X\mathcal{Q}}\|_{B_0}. \tag{6.32}$$

where $\tau_{X\mathcal{Q}}(\mu)$ denotes the individual parametric error estimate associated with the index $\mu \in \mathcal{Q}$. The total two-level error estimate will be denoted by

$$\tau^2_{X\mathcal{P}} := \tau_{X\mathcal{P}}(\mathcal{N}^+, \mathcal{Q})^2 := \tau_{Y\mathcal{P}}(\mathcal{N}^+)^2 + \tau_{X\mathcal{Q}}(\mathcal{Q})^2. \tag{6.33}$$

Furthermore, for any subset of midpoints $\mathcal{M} \subseteq \mathcal{N}^+$ and indices $\mathcal{M} \subseteq \mathcal{Q}$, the following notation naturally follows:

$$\tau_{Y\mathcal{P}}(\mathcal{M})^2 := \sum_{\boldsymbol{z}\in\mathcal{M}} \tau_{Y\mathcal{P}}(\boldsymbol{z})^2, \ \tau_{X\mathcal{Q}}(\mathcal{M})^2 := \sum_{\mu\in\mathcal{M}} \tau_{X\mathcal{Q}}(\mu)^2, \text{ and } \tau_{X\mathcal{P}}(\mathcal{M},\mathcal{M})^2 := \tau_{Y\mathcal{P}}(\mathcal{M})^2 + \tau_{X\mathcal{Q}}(\mathcal{M})^2. \tag{6.34}$$

Remark 6.2.1. *Notation* (6.31) *for local spatial error estimates is not customary since the indexing is done using midpoints instead of edges. We stress, however, that it is perfectly equivalent to consider*

midpoints instead of their associated edges as well as considering sets of marked midpoints $\mathcal{M} \subseteq \mathcal{N}^+$ (see (6.34)*) rather than sets of marked edges $\mathcal{M} \subseteq \mathcal{E}(\mathcal{T})$.*

Remark 6.2.2. *The local error estimate $\tau_{X\mathcal{Q}}(\mu)$ in* (6.32) *coincides with $\eta_{X\mathcal{Q}}(\mu)$ defined in* (5.38)*, since, as already noticed, the parametric parts of hierarchical estimate $\eta_{X\mathcal{P}}$ and two-level estimate $\tau_{X\mathcal{P}}$ are the same. However, for consistency of notation, we change the definition of the single estimate $\| e^{(\mu)}_{X\mathcal{Q}} \|_{B_0}$, $\mu \in \mathcal{Q}$, using either the symbol η or τ according to the error estimate we refer to.*

6.2.2 Schematic adaptive loop

In the same spirit of adaptive Algorithm 5.1, we now introduce an adaptive algorithm driven by two-level estimates for parametric problem (4.5). We present the algorithm in a schematic way (or abstract form) by minimising dependence on data and with no specification of working subroutines. Hereafter, we use the iteration counter $\ell \in \mathbb{N}_0$ of the loop of the adaptive algorithm to denote triangulations, index sets, Galerkin solutions, etc., associated with the ℓ-th iteration of the loop.

Let $\mathcal{T}_0$ be the underlying initial conforming coarse triangulation of the domain D and let $\mathcal{P}_0$ be the initial index set. For each $\ell \in \mathbb{N}_0$, iterations of the algorithm read as follows. The discrete Galerkin approximation $u_\ell \in V_\ell$ satisfying (4.17) is computed. The total two-level error estimate $\tau_\ell := \tau^{(\ell)}_{X\mathcal{P}}$ is assembled as in (6.33) by computing local spatial $\tau_\ell(z) := \tau^{(\ell)}_{Y\mathcal{P}}(z)$ and parametric $\tau_\ell(\mu) := \tau^{(\ell)}_{X\mathcal{Q}}(\mu)$ estimates for all $z \in \mathcal{N}^+_\ell$ and $\mu \in \mathcal{Q}_\ell$. Here, $\mathcal{Q}_\ell$ is the detail index set defined in (5.35). A marking criterion returns two subsets $\mathcal{M}_\ell \subseteq \mathcal{N}^+_\ell$ and $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$ and finally the algorithm sets $\mathcal{P}_{\ell+1} = \mathcal{P}_\ell \cup \mathcal{M}_\ell$ and $\mathcal{T}_{\ell+1} = \mathsf{REFINE}(\mathcal{T}_\ell, \mathcal{M}_\ell)$, where $\mathcal{T}_{\ell+1}$ is obtained by local NVB refinements with reference edges to be the longest edges of each element of $\mathcal{T}_\ell$.

The adaptive algorithm generates a sequence $(\mathcal{T}_\ell)_{\ell \in \mathbb{N}_0}$ of adaptively refined triangulations and a sequence $(\mathcal{P}_\ell)_{\ell \in \mathbb{N}_0}$ of adaptively enriched index sets such that, for all $\ell \in \mathbb{N}_0$, there holds

$$X_\ell \subseteq X_{\ell+1} \subseteq \widehat{X}_\ell \subset H^1_0(D) \quad \text{and} \quad \mathcal{P}_{\mathcal{P}_\ell} \subseteq \mathcal{P}_{\mathcal{P}_{\ell+1}} \subseteq \widehat{\mathcal{P}}_{\mathcal{P}_\ell} \subseteq \widehat{\mathcal{P}}_{\mathcal{P}_{\ell+1}} \subset L^2_\pi(\Gamma),$$

where recall that $\widehat{\mathcal{P}}_{\mathcal{P}_\ell} = \mathcal{P}_{\mathcal{P}_\ell} \oplus \mathcal{P}_{\mathcal{Q}_\ell}$. Furthermore, the algorithm performs either a mesh-refinement or a parametric enrichment at each iteration, and thus, for $\ell \in \mathbb{N}_0$, one of the inclusions $X_\ell \subseteq X_{\ell+1}$ or $\mathcal{P}_{\mathcal{P}_\ell} \subseteq \mathcal{P}_{\mathcal{P}_{\ell+1}}$ is strict (in other words, at each iteration, either $\mathcal{M}_\ell$ or $\mathcal{M}_\ell$ is empty). Therefore, we have $V_\ell \subset V_{\ell+1}$ as well as $\widehat{V}_\ell \subset \widehat{V}_{\ell+1}$. In particular, the enrichment type depends on the marking criterion used (this is specified subsequently). The adaptive algorithm is listed in Algorithm 6.1.

Schematic adaptive SGFEM algorithm

Input: *triangulation* $\mathcal{T}_0$, *index set* $\mathcal{P}_0$;
Set $\ell = 0$;

(i) Compute the discrete approximation $u_\ell \in V_\ell$;

(ii) Assemble the total error estimate τ_ℓ *by computing* $\tau_\ell(z)$ *and* $\tau_\ell(\mu)$ *for all* $z \in \mathcal{N}_\ell^+$ *and all* $\mu \in \mathcal{Q}_\ell$;

(iii) Use a marking criterion to select a subset $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ *and a subset* $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$;

(iv) Set $\mathcal{P}_{\ell+1} := \mathcal{P}_\ell \cup \mathcal{M}_\ell$ *and* $\mathcal{T}_{\ell+1} = \mathsf{REFINE}(\mathcal{T}_\ell, \mathcal{M}_\ell)$;

(v) Increase the counter $\ell \mapsto \ell + 1$ *and go back to (i).*

Output: *sequence of triangulations* $\mathcal{T}_\ell$, *index sets* $\mathcal{P}_\ell$, *Galerkin solutions* u_ℓ, *and error estimates* τ_ℓ.

Algorithm 6.1. Schematic adaptive SGFEM algorithm driven by two-level error estimates for parametric problem (4.5).

Note that after step *(ii)*, a stopping criterion is used, in practice, to terminate the algorithm (cf. Algorithm 5.1).

6.2.3 Estimates of the error reduction

Let us emphasise that the analysis of the two-level error estimate made in Section 6.1 proves a more general result than error reduction (6.6) for the pair of Galerkin solutions $u_\ell \in V_\ell$ and $\widehat{u}_\ell \in \widehat{V}_\ell$. The proof of Theorem 6.1 essentially relies on the stable subspace decompositions

$$\widehat{X}_\ell = X_\ell \oplus Y_\ell = X_\ell \oplus \left(\bigoplus_{z \in \mathcal{N}_\ell^+} \mathsf{span}\{\psi_{z,\ell}\} \right),$$

$$\widehat{\mathcal{P}}_\mathcal{P} = \mathcal{P}_{\mathcal{P}_\ell} \oplus \mathcal{P}_{\mathcal{Q}_\ell} = \mathcal{P}_{\mathcal{P}_\ell} \oplus \left(\bigoplus_{\mu \in \mathcal{Q}_\ell} \mathsf{span}\{P_{\mu,\ell}\} \right).$$

In particular, let $\mathcal{T}_{\ell+1}$ be the refined triangulation returned by the REFINE subroutine for a given set of marked midpoints $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$. Consider a midpoint $z \in \mathcal{R}_\ell \subseteq \mathcal{N}_\ell^+$, where $\mathcal{R}_\ell \supseteq \mathcal{M}_\ell$ is the set of *refined midpoints* given by $\mathcal{R}_\ell = \mathcal{N}_{\ell+1}^\circ \cap \mathcal{N}_\ell^+ = \mathcal{N}_{\ell+1}^\circ \setminus \mathcal{N}_\ell^\circ$, i.e., the set consisting of $\mathcal{M}_\ell$ and all extra midpoints that are marked by completion steps of the NVB refinements to keep the conformity of $\mathcal{T}_{\ell+1}$ (see Figure 6.1). Let $\varphi_{z,\ell+1} \in X_{\ell+1}$ be the corresponding piecewise linear hat function associated with z on the (locally) refined triangulation $\mathcal{T}_{\ell+1}$. Also, let $\widehat{\varphi}_{z,\ell} \in \widehat{X}_\ell$ be the corresponding piecewise linear hat function associated with z on the (uniformly) refined triangulation $\widehat{\mathcal{T}}_\ell$. The refinement made by NVB ensures that $\psi_{z,\ell} = \varphi_{z,\ell+1} = \widehat{\varphi}_{z,\ell}$, and this, in turn, yields the stable

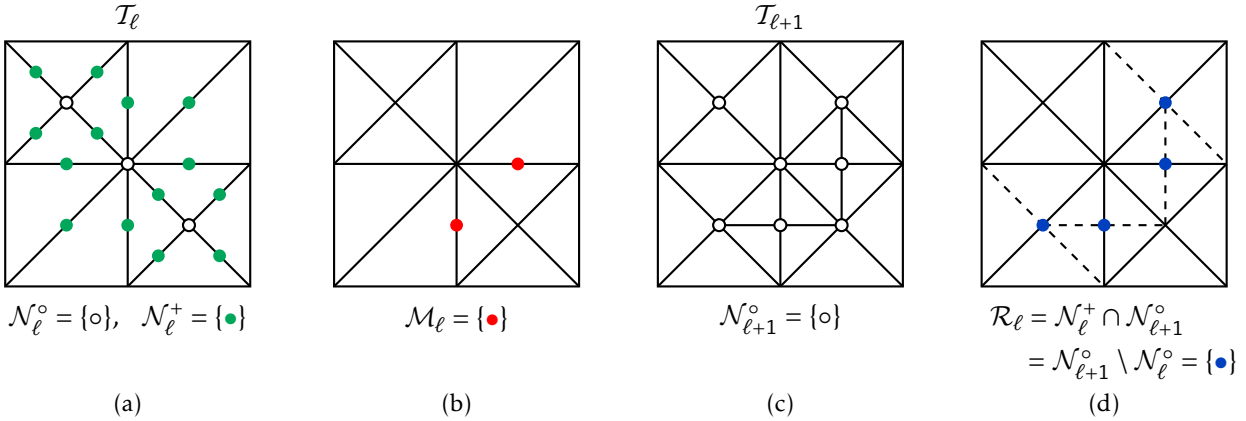


Figure 6.1. (a) Triangulation $\mathcal{T}_\ell$ with interior vertices $\mathcal{N}_\ell^\circ$ (white dots) and midpoints $\mathcal{N}_\ell^+$ of all interior edges (green dots); (b) Set $\mathcal{M}_\ell$ of marked midpoints (red dots); (c) Conforming triangulation $\mathcal{T}_{\ell+1}$ with interior vertices $\mathcal{N}_{\ell+1}^\circ$ (white dots); (d) Set of refined midpoints $\mathcal{R}_\ell$ (blue dots) that are introduced by the refinement.

decomposition

$$X_{\ell+1} = X_\ell \oplus \left(\bigoplus_{z \in \mathcal{R}_\ell} \operatorname{span}\{\varphi_{z,\ell+1}\} \right) = X_\ell \oplus \left(\bigoplus_{z \in \mathcal{R}_\ell} \operatorname{span}\{\widehat{\varphi}_{z,\ell}\} \right).$$

We emphasise that this property does not hold for other mesh-refinement techniques which do not lead, in general, to nested finite element spaces (see Remark 2.3.1). Therefore, the following result shows how the spatial and parametric contributing part to the total two-level estimate τ_ℓ can be used to control the error reduction due to adaptive enrichments of the components of the approximation space $V_\ell = X_\ell \otimes \mathcal{P}_{\mathcal{P}_\ell}$.

Corollary 6.1. *Let $C_{thm} \geq 1$ be the constant from Theorem 6.1. Let $\widehat{\mathcal{T}_\ell} = \mathsf{REFINE}(\mathcal{T}_\ell, \mathcal{N}_\ell^+)$ be the uniform triangulation obtained by NVB refinements. Analogously, suppose that $\mathcal{T}_{\ell+1} = \mathsf{REFINE}(\mathcal{T}_\ell, \mathcal{M}_\ell)$ is the locally refined triangulation obtained by NVB refinements for a subset of midpoints $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$. Also, let $\mathcal{P}_{\ell+1} = \mathcal{P}_\ell \cup \mathcal{M}_\ell$ for a subset of indices $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$. If $u_\ell \in V_\ell$ and $u_{\ell+1} \in V_{\ell+1}$ the are corresponding Galerkin approximations, then there holds*

$$\frac{\lambda}{K} \tau_\ell\big(\mathcal{R}_\ell, \mathcal{M}_\ell\big)^2 \leq \| u_{\ell+1} - u_\ell \|_B^2 \leq \Lambda C_{thm} \tau_\ell\big(\mathcal{R}_\ell, \mathcal{M}_\ell\big)^2. \tag{6.35}$$

6.2.4 Marking criteria

We now describe four different marking criteria, to be used in Step *(iii)* of Algorithm 6.1, that specify the selection of the subsets $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ and $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$ and, at the same time, determine the type of enrichment of the ℓ-th iteration of the adaptive loop. In particular, each criterion comes

Marking criterion for adaptive SGFEM

Input: *error estimates* $\{\tau_\ell(z)\}_{z\in\mathcal{N}_\ell^+}$, $\{\tau_\ell(\mu)\}_{\mu\in\mathcal{Q}_\ell}$, $\vartheta > 0$, *and marking parameters* $\theta_X, \theta_{\mathcal{P}} \in (0,1]$.
Case (a): $\tau_\ell(\mathcal{N}_\ell^+) \geq \vartheta\,\tau_\ell(\mathcal{Q}_\ell)$

set $\mathcal{M}_\ell := \emptyset$;

find $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ *with minimal cardinality such that* $\tau_\ell(\mathcal{M}_\ell) \geq \theta_X \tau_\ell(\mathcal{N}_\ell^+)$.

Case (b): $\tau_\ell(\mathcal{N}_\ell^+) < \vartheta\,\tau_\ell(\mathcal{Q}_\ell)$

set $\mathcal{M}_\ell := \emptyset$;

find $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$ *with minimal cardinality such that* $\tau_\ell(\mathcal{M}_\ell) \geq \theta_{\mathcal{P}} \tau_\ell(\mathcal{Q}_\ell)$.

Output: $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ *and* $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$, *where one of the two subsets is empty.*

Criterion 6.1. A marking criterion based on total error estimates for an adaptive SGFEM algorithm driven by two-level estimates.

Marking criterion for adaptive SGFEM

Input: *error estimates* $\{\tau_\ell(z)\}_{z\in\mathcal{N}_\ell^+}$, $\{\tau_\ell(\mu)\}_{\mu\in\mathcal{Q}_\ell}$, $\vartheta > 0$, *and marking parameters* $\theta_X, \theta_{\mathcal{P}} \in (0,1]$.

find $\widetilde{\mathcal{M}}_\ell \subseteq \mathcal{Q}_\ell$ *with minimal cardinality such that* $\tau_\ell(\widetilde{\mathcal{M}}_\ell) \geq \theta_{\mathcal{P}}\,\tau_\ell(\mathcal{Q}_\ell)$;
find $\widetilde{\mathcal{M}}_\ell \subseteq \mathcal{N}_\ell^+$ *with minimal cardinality such that* $\tau_\ell(\widetilde{\mathcal{M}}_\ell) \geq \theta_X\,\tau_\ell(\mathcal{N}_\ell^+)$;
set $\widetilde{\mathcal{R}}_\ell := \mathcal{N}_{\ell+1}^\circ \cap \mathcal{N}_\ell^+$, *where* $\mathcal{N}_{\ell+1}^\circ$ *is associated with* $\mathcal{T}_{\ell+1} = \mathsf{REFINE}(\mathcal{T}_\ell, \widetilde{\mathcal{M}}_\ell)$.

Case (a): $\tau_\ell(\widetilde{\mathcal{R}}_\ell) \geq \vartheta\,\tau_\ell(\widetilde{\mathcal{M}}_\ell)$. *Set* $\mathcal{M}_\ell = \emptyset$ *and* $\mathcal{M}_\ell = \widetilde{\mathcal{M}}_\ell$.
Case (b): $\tau_\ell(\widetilde{\mathcal{R}}_\ell) < \vartheta\,\tau_\ell(\widetilde{\mathcal{M}}_\ell)$. *Set* $\mathcal{M}_\ell = \widetilde{\mathcal{M}}_\ell$ *and* $\mathcal{M}_\ell = \emptyset$.

Output: $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ *and* $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$, *where one of the two subsets is empty.*

Criterion 6.2. A marking criterion based on error reduction estimates for an adaptive SGFEM algorithm driven by two-level estimates.

with three parameters: $\vartheta > 0$ is a weight modulating the choice between mesh refinement and parametric enrichment (with parametric enrichment being favoured for $\vartheta > 1$) and θ_X, $\theta_{\mathcal{P}} \in (0,1]$ are the spatial and parametric threshold parameters controlling the marking of midpoints in $\mathcal{N}_\ell^+$ and the marking of indices in $\mathcal{Q}_\ell$, respectively.

The first criterion is similar to Criterion 5.1 used by adaptive Algorithm 5.1. That is, it enforces spatial refinement if the spatial error estimate is comparably large; otherwise, parametric enrichment is chosen for the next iteration. Marked nodes (resp., marked indices) are obtained via Dörfler marking (see Remark 6.2.3). This criterion is listed in Criterion 6.1.

Similarly to the first one, the second criterion is inspired by Criterion 5.2 used by adaptive Algorithm 5.1. That is, it is based on the idea that the error estimate τ_ℓ based on the refined

Marking criterion for adaptive SGFEM

Input: *error estimates* $\{\tau_\ell(z)\}_{z\in\mathcal{N}_\ell^+}$, $\{\tau_\ell(\mu)\}_{\mu\in\mathcal{Q}_\ell}$, $\vartheta > 0$, *and marking parameters* $\theta_X, \theta_{\mathcal{P}} \in (0,1]$.
Case (a): $\tau_\ell(\mathcal{N}_\ell^+) \geq \vartheta\,\tau_\ell(\mathcal{Q}_\ell)$

set $\mathcal{M}_\ell := \emptyset$;

find $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ *with minimal cardinality such that* $\tau_\ell(\mathcal{M}_\ell) \geq \theta_X \tau_\ell(\mathcal{N}_\ell^+)$.

Case (b): $\tau_\ell(\mathcal{N}_\ell^+) < \vartheta\,\tau_\ell(\mathcal{Q}_\ell)$

set $\mathcal{M}_\ell := \emptyset$;

define $\mathcal{M}_\ell := \{\mu \in \mathcal{Q}_\ell \,:\, \tau_\ell(\mu) \geq (1-\theta_{\mathcal{P}})\max_{\mu\in\mathcal{Q}_\ell}\tau_\ell(\mu)\}$.

Output: $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ *and* $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$, *where one of the two subsets is empty.*

Criterion 6.3. A marking criterion based on total error estimates for an adaptive SGFEM algorithm driven by two-level estimates.

Marking criterion for adaptive SGFEM

Input: *error estimates* $\{\tau_\ell(z)\}_{z\in\mathcal{N}_\ell^+}$, $\{\tau_\ell(\mu)\}_{\mu\in\mathcal{Q}_\ell}$, $\vartheta > 0$, *and marking parameters* $\theta_X, \theta_{\mathcal{P}} \in (0,1]$.

define $\widetilde{\mathcal{M}}_\ell := \{\mu \in \mathcal{Q}_\ell \,:\, \tau_\ell(\mu) \geq (1-\theta_{\mathcal{P}})\max_{\mu\in\mathcal{Q}_\ell}\tau_\ell(\mu)\}$;

find $\widetilde{\mathcal{M}}_\ell \subseteq \mathcal{N}_\ell^+$ *with minimal cardinality such that* $\tau_\ell(\widetilde{\mathcal{M}}_\ell) \geq \theta_X\,\tau_\ell(\mathcal{N}_\ell^+)$;

set $\widetilde{\mathcal{R}}_\ell := \mathcal{N}_{\ell+1}^\circ \cap \mathcal{N}_\ell^+$, *where* $\mathcal{N}_{\ell+1}^\circ$ *is associated with* $\mathcal{T}_{\ell+1} = \mathsf{REFINE}(\mathcal{T}_\ell, \widetilde{\mathcal{M}}_\ell)$.

Case (a): $\tau_\ell(\widetilde{\mathcal{R}}_\ell) \geq \vartheta\,\tau_\ell(\widetilde{\mathcal{M}}_\ell)$. *Set* $\mathcal{M}_\ell = \emptyset$ *and* $\mathcal{M}_\ell = \widetilde{\mathcal{M}}_\ell$.
Case (b): $\tau_\ell(\widetilde{\mathcal{R}}_\ell) < \vartheta\,\tau_\ell(\widetilde{\mathcal{M}}_\ell)$. *Set* $\mathcal{M}_\ell = \widetilde{\mathcal{M}}_\ell$ *and* $\mathcal{M}_\ell = \emptyset$.

Output: $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ *and* $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$, *where one of the two subsets is empty.*

Criterion 6.4. A marking criterion based on error reduction estimates for an adaptive SGFEM algorithm driven by two-level estimates.

midpoints (resp., added indices) provides information about the associated error reduction (see Corollary 6.1). This criterion, which is listed in Criterion 6.2, enforces either spatial refinements (if the error reduction for spatial mesh-refinement is comparably large) or parametric enrichments (otherwise).

Criterion 6.3 is a modification of Criterion 6.1. It employs a maximum strategy (see Remark 6.2.3) in the parameter domain, while using the Dörfler strategy in the physical domain. As in Criterion 6.1, the enrichment type is determined by the dominant contributing error estimate.

Finally, Criterion 6.4 is a modification of Criterion 6.2 in the same way as Criterion 6.3 is a modification of Criterion 6.1. Namely, we employ the Dörfler strategy in the physical domain

and use a maximum strategy in the parameter domain, while the refinement type for successive iterations is determined by the dominant error reduction.

Remark 6.2.3 (Dörfler and maximum strategies)**.** *In all proposed Criteria 6.1–6.4, the selection of marked midpoints $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ for spatial refinements occur via the Dörfler strategy. However, the criteria use a modified version of the Dörfler strategy described in Strategy 2.2 (cf.* (2.15) *with the condition satisfied by marked midpoints in, e.g., Criterion 6.1). From the algorithmic point of view, numerical experiments show that the Dörfler strategies as used in Criteria 6.1–6.4 effectively mark less 'elements' per iteration than those marked by Strategy 2.2.*

For what concerns Criteria 6.3 and 6.4, the maximum strategy employed here is also a modification of Strategy 2.1 (cf. (2.14) *with the condition satisfied by marked indices in, e.g., Criterion 6.3). The modification is only the input threshold $(1-\theta_{\mathcal{P}})$ for the marking strategy. In particular, notice that this way, large values of marking parameter $\theta_{\mathcal{P}}$ lead to large subsets of marked indices and vice versa.*

Finally, notice that for $\theta_{\mathcal{X}} = \theta_{\mathcal{P}} = 1$, Criteria 6.1–6.4 return the same full subsets of marked midpoints $\mathcal{M}_\ell = \mathcal{N}_\ell^+$ and marked indices $\mathcal{M}_\ell = \mathcal{Q}_\ell$.

In the reminder of the chapter we will write Algorithm 6.1v1 to refer to the version of adaptive Algorithm 6.1 which employs Criterion 6.1 in its Step *(iii)*; similarly, for other choices of criteria. When we refer to Algorithm 6.1 without specifying the version type, this will mean that the statement holds for any of the four proposed marking criteria.

6.3 Analysis of convergence of the adaptive algorithm

In this section, we report the main results about the convergence analysis of Algorithm 6.1 for parametric model problem (4.5). In particular, we start by stating the *plain* convergence of the generated sequence of two-level error estimates, which holds for the proposed four versions of Algorithm 6.1. This result follows from the convergence analysis in [25]. Then, we prove that under saturation assumption (5.14), the versions of the algorithm using the Dörfler strategy to mark both spatial and parametric components of discretisation error in the corresponding marking criterion (i.e., Algorithms 6.1v1 and 6.1v2) yield a sequence of global energy errors which converge *linearly*.

6.3.1 Convergence results

The first convergence result shows that Algorithm 6.1 ensures convergence of the computed sequence of two-level error estimates to zero.

Theorem 6.2 (Plain convergence). *For any choice of marking parameters θ_X, $\theta_{\mathcal{P}}$, and ϑ, Algorithm 6.1 yields a convergent sequence of error estimates, i.e., $\tau_\ell \to 0$ as $\ell \to \infty$.*

We emphasise that this result is valid independently of saturation assumption (5.14) and no additional assumptions on the refinement level of the underlying sequence of triangulations is required (cf. [55]). For the proof of Theorem 6.2, see Sections 6 and 7 in [25].

Remark 6.3.1. *Note that although proposed Criteria 6.1–6.4 seem to be the natural candidates in our present setting, the proof of Theorem 6.2 allows for more general marking criteria than those proposed in Section 6.2.1 (see Propositions 10 and 11 in [25]).*

The following result is an immediate consequence of Theorem 6.2 and upper bound (6.7) from Theorem 6.1. That is, trivially, the convergence of two-level estimates and their global reliability (under saturation assumption (5.14)) ensure the converge of the sequence of energy errors to zero.

Corollary 6.2. *Let $u \in V$ be the solution to problem* (4.17). *Let $(u_\ell)_{\ell\in\mathbb{N}_0}$ be the sequence of Galerkin solutions generated by Algorithm 6.1. Denote by $(\widehat{u}_\ell)_{\ell\in\mathbb{N}_0}$ the associated sequence of Galerkin solutions satisfying* (5.13) *and suppose that saturation assumption* (5.14) *holds for each pair u_ℓ and $\widehat{u}_\ell$ ($\ell \in \mathbb{N}_0$). Then, for any choice of marking parameters θ_X, $\theta_{\mathcal{P}}$, and ϑ, Algorithm 6.1 yields convergence of the energy norm of the error, i.e., $\|u - u_\ell\|_B \to 0$ as $\ell \to \infty$.*

Under saturation assumption (5.14), Algorithms 6.1v1 and 6.1v2 allow for a stronger convergence result than Corollary 6.2. The following theorem states the *linear* convergence of the computed sequence of energy errors. The proof of this result is given in the next section.

Theorem 6.3 (Linear convergence). *Let $u \in V$ be the solution to problem* (4.17). *Let $(u_\ell)_{\ell\in\mathbb{N}_0}$ be the sequence of Galerkin solutions generated by either Algorithm 6.1v1 or Algorithm 6.1v2 with arbitrary $\theta_X, \theta_{\mathcal{P}} \in (0,1]$ and $\vartheta > 0$. Denote by $(\widehat{u}_\ell)_{\ell\in\mathbb{N}_0}$ the associated sequence of Galerkin solutions satisfying* (5.13) *and suppose that saturation assumption* (5.14) *holds for each pair u_ℓ and $\widehat{u}_\ell$ ($\ell \in \mathbb{N}_0$). Then, there exists a positive constant $q_{\mathrm{lin}} < 1$ such that*

$$\|u - u_{\ell+1}\|_B \;\leq\; q_{\mathrm{lin}}\,\|u - u_\ell\|_B \qquad \forall\, \ell \in \mathbb{N}_0 .$$

The constant q_{lin} depends on the mean field a_0 in (4.8)*, the constant γ in* (4.10)*, the saturation constant q_{sat} in* (5.14)*, the coarse triangulation $\mathcal{T}_0$, and marking parameters θ_X, $\theta_{\mathcal{P}}$, and ϑ.*

6.3.2 Linear convergence of the energy errors

In this section, we prove that saturation assumption (5.14) yields contraction of the energy errors at each iteration of Algorithms 6.1v1 and 6.1v2. In the proof, we adapt the arguments of [96].

Firstly, let us prove, in the following lemma, the contraction of the energy errors for all those iterations of the algorithms where spatial refinements are performed.

Lemma 6.5. *Let $u \in V$ be the solution to problem* (4.17)*. Let $\ell \in \mathbb{N}_0$ and suppose that saturation assumption* (5.14) *holds for two Galerkin solutions u_ℓ and $\widehat{u}_\ell$ satisfying* (4.36) *and* (5.10)*, respectively. Suppose that*

$$\tau_\ell(\mathcal{Q}_\ell) \leq C_\vartheta\, \tau_\ell(\mathcal{N}_\ell^+) \quad \text{with } C_\vartheta > 0, \tag{6.36}$$

and let $\mathcal{M}_\ell \subseteq \mathcal{N}_{\ell+1}^\circ \cap \mathcal{N}_\ell^+ = \mathcal{R}_\ell$ satisfy

$$\tau_\ell(\mathcal{M}_\ell) \geq \theta\, \tau_\ell(\mathcal{N}_\ell^+) \quad \text{with } \theta \in (0,1]. \tag{6.37}$$

Then, for the enhanced Galerkin solution $u_{\ell+1} \in X_{\ell+1} \otimes \mathcal{P}_{\mathcal{P}_\ell}$, there holds

$$\|u - u_{\ell+1}\|_B^2 \leq (1-q)\|u - u_\ell\|_B^2,$$

where $q \in (0,1)$ depends on the mean field a_0 in (4.8)*, the initial triangulation $\mathcal{T}_0$, the constant γ in* (4.10)*, the saturation constant q_{sat} in* (5.14)*, C_ϑ, and θ.*

Proof. Using the upper bound in (6.7), inequality (6.36), and marking strategy (6.37), we obtain

$$\begin{aligned}\frac{1-q_{\mathrm{sat}}^2}{\Lambda C_{\mathrm{thm}}}\|u - u_\ell\|_B^2 &\overset{(6.7)}{\leq} \tau_\ell \overset{(6.33)}{=} \tau_\ell(\mathcal{N}_\ell^+)^2 + \tau_\ell(\mathcal{Q}_\ell)^2 \overset{(6.36)}{\leq} (1 + C_\vartheta^2)\,\tau_\ell(\mathcal{N}_\ell^+)^2 \\ &\overset{(6.37)}{\leq} (1 + C_\vartheta^2)\theta^{-2}\,\tau_\ell(\mathcal{M}_\ell)^2.\end{aligned} \tag{6.38}$$

Hence, using Corollary 6.1 and the fact that $\mathcal{M}_\ell \subseteq \mathcal{R}_\ell$, we derive that

$$\begin{aligned}\|u - u_{\ell+1}\|_B^2 &\overset{(5.13)}{=} \|u - u_\ell\|_B^2 - \|u_{\ell+1} - u_\ell\|_B^2 \\ &\overset{(6.35)}{\leq} \|u - u_\ell\|_B^2 - \frac{\lambda}{K}\tau_\ell(\mathcal{R}_\ell, \emptyset)^2 \\ &\leq \|u - u_\ell\|_B^2 - \frac{\lambda}{K}\tau_\ell(\mathcal{M}_\ell, \emptyset)^2 \overset{(6.38)}{\leq} \Bigg(1 - \underbrace{\frac{\lambda\,\theta^2(1-q_{\mathrm{sat}}^2)}{\Lambda\, C_{\mathrm{thm}}(1+C_\vartheta^2)K}}_{=:q}\Bigg)\|u - u_\ell\|_B^2.\end{aligned}$$

This concludes the proof. □

Similarly to Lemma 6.5, the next result concerns the contraction of the energy errors for those iterations of Algorithms 6.1v1 and 6.1v2 where parametric enrichments are performed. The proof follows from the same arguments used in Lemma 6.5, so it is omitted.

Lemma 6.6. *Let $u \in V$ be the solution to problem* (4.17)*. Let $\ell \in \mathbb{N}_0$ and suppose that the saturation assumption* (5.14) *holds for two Galerkin solutions u_ℓ and $\widehat{u}_\ell$ satisfying* (4.36) *and* (5.10)*, respectively. Suppose that*

$$\tau_\ell(\mathcal{N}_\ell^+) \leq C_\vartheta \, \tau_\ell(\mathcal{Q}_\ell) \quad \text{with } C_\vartheta > 0,$$

and let $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell \cap \mathcal{P}_{\ell+1}$ be such that

$$\theta \, \tau_\ell(\mathcal{Q}_\ell) \leq \tau_\ell(\mathcal{M}_\ell) \quad \text{with } \theta \in (0,1].$$

Then, for the enhanced Galerkin solution $u_{\ell+1} \in X_\ell \otimes \mathcal{P}_{\mathcal{P}_{\ell+1}}$, there holds

$$\| u - u_{\ell+1} \|_B^2 \leq (1-q) \| u - u_\ell \|_B^2,$$

where $q \in (0,1)$ depends on the mean field a_0 in (4.8)*, the initial triangulation $\mathcal{T}_0$, the constant γ in* (4.10)*, the saturation constant q_{sat} in* (5.14)*, C_ϑ, and θ.*

With the previous two lemmas, we are now ready to prove Theorem 6.3.

Proof of Theorem 6.3. Firstly, consider Algorithm 6.1v1. If *Case (a)* of marking Criterion 6.1 occurs, we can apply Lemma 6.5 with $C_\vartheta = \vartheta^{-1}$ and $\theta = \theta_X$. Likewise, when *Case (b)* (of the same marking criterion) occurs, we can use Lemma 6.6 with $C_\vartheta = \vartheta$ and $\theta = \theta_{\mathcal{P}}$. In both cases, this proves the contraction of the energy error for a constant $q_{\mathsf{lin}} \in (0,1)$, i.e., there holds $\| u - u_{\ell+1} \|_B \leq q_{\mathsf{lin}} \| u - u_\ell \|_B$ for all $\ell \in \mathbb{N}_0$.

Let us now consider Algorithm 6.1v2. In *Case (a)* of marking Criterion 6.2 one has

$$\theta_{\mathcal{P}} \, \tau_\ell(\mathcal{Q}_\ell) \leq \tau_\ell(\widetilde{\mathcal{M}}_\ell) \leq \vartheta^{-1} \tau_\ell(\widetilde{\mathcal{R}}_\ell) \leq \vartheta^{-1} \tau_\ell(\mathcal{N}_\ell^+).$$

Hence, Lemma 6.5 applies to this case with $C_\vartheta = \theta_{\mathcal{P}}^{-1} \vartheta^{-1}$ and $\theta = \theta_X$. Similarly, in *Case (b)* of marking Criterion 6.2, one has

$$\theta_X \, \tau_\ell(\mathcal{N}_\ell^+) \leq \tau_\ell(\widetilde{\mathcal{M}}_\ell) \leq \tau_\ell(\widetilde{\mathcal{R}}_\ell) < \vartheta \, \tau_\ell(\widetilde{\mathcal{M}}_\ell) \leq \vartheta \, \tau_\ell(\mathcal{Q}_\ell),$$

and thus, in this case, Lemma 6.6 applies with $C_\vartheta = \theta_X^{-1} \vartheta$ and $\theta = \theta_{\mathcal{P}}$. In both cases, we therefore obtain the contraction of the energy error, $\| u - u_{\ell+1} \|_B \leq q_{\mathsf{lin}} \| u - u_\ell \|_B$ for all $\ell \in \mathbb{N}_0$, for a constant $q_{\mathsf{lin}} \in (0,1)$. □

6.4 Numerical experiments

In this section, we report the results of two numerical experiments performed for the parametric model problem (4.5). In the first experiment, we compare the performance of adaptive Algorithm 6.1 driven by two-level error estimates (see (6.5)) with Algorithm 5.1 driven by hierarchical error estimates (see (5.22)). The second experiment is focused on comparing the performance of Algorithms 6.1v1–6.1v4 for a certain range of marking parameters. The experiments were performed using the toolbox Stochastic T-IFISS [28] (see Appendix B) on a desktop computer equipped with an Intel Core CPU i5-4590@3.30GHz and 8.00GB of RAM.

6.4.1 Experiment 1 - Comparison with hierarchical estimates

Consider the parametric model problem, posed on the square domain $D = (0,1)^2$, described in the numerical experiment of Section 5.4.2. Our aim is to show the advantages in running adaptive algorithms driven by two-level error estimates rather than hierarchical error estimates. To this end, we focus the attention only on the versions of the algorithms which use similar marking criteria, i.e., Algorithms 5.1v1 and 6.1v1 and Algorithms 5.1v2 and 6.1v2.

Recall from Remark 6.2.3 that Algorithm 6.1 (with its four possible marking Criteria 6.1–6.4) employs a modified version of the Dörfler strategy that is used by Algorithm 5.1. Thus, in order to make a meaningful comparison, we need to adapt Algorithms 6.1v1 and 6.1v2 appropriately. That is, for this experiment, we consider the modification of Criterion 6.1 such that the subset of marked midpoints $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ satisfies $\tau_\ell(\mathcal{M}_\ell)^2 \geq \theta_X \tau_\ell(\mathcal{N}_\ell^+)^2$ for iterations $\ell \in \mathbb{N}_0$ (cf. (2.15)); the same amendment is done for the subset $\widetilde{\mathcal{M}}_\ell \subseteq \mathcal{N}_\ell^+$ in Criterion 6.2.

We use the initial coarse triangulation $\mathcal{T}_0$ depicted in Figure 5.3(a) and the initial index set $\mathcal{P}_0$ defined in (5.42). Detail index sets $\mathcal{Q}_\ell$ are constructed via (5.35) with $M_{\mathcal{Q}} = 1$. In addition, we set $\vartheta = 1$ in Criteria 6.1 and 6.2. We first run adaptive Algorithms 5.1v1 and 6.1v1 with marking parameters $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.8$, and then adaptive Algorithms 5.1v2 and 6.1v2 with marking parameters $\theta_X = 0.25$, $\theta_{\mathcal{P}} = 0.8$, for the prescribed tolerance tol = 1.0e-3; see Figures 5.3(c) and 5.3(d) for a plot of the computed SGFEM mean and variance for this model problem.

Table 6.1 reports the results of the computations, including the same set of data as described in Section 5.4.1. Notice that the last computed hierarchical estimate η_L refers to Algorithms 5.1v1 and 5.1v2 while the last computed two-level estimate τ_L refers to Algorithms 6.1v1 and 6.1v2. For

	$\theta_X = 0.5,\ \theta_{\mathcal{P}} = 0.8$		$\theta_X = 0.25,\ \theta_{\mathcal{P}} = 0.8$	
	Algorithm 5.1v1	Algorithm 6.1v1	Algorithm 5.1v2	Algorithm 6.1v2
L	31	23	56	44
t (sec)	940	218	1066	362
$\eta_L(\tau_L)$	9.2984e-04	9.9701e-04	9.9668e-04	9.8483e-04
N_L	2,649,625	1,310,575	2,103,342	1,555,772
$\#\mathcal{T}_L$	213,208	105,688	124,648	92,294
$\#\mathcal{P}_L$	25	25	34	34
$M_{\mathcal{P}_L}$	7	7	8	8

Table 6.1. The results of running Algorithms 5.1v1 and 6.1v1 with $\theta_X = 0.5$ and $\theta_{\mathcal{P}} = 0.8$ and Algorithms 5.1v2 and 6.1v2 with $\theta_X = 0.25$ and $\theta_{\mathcal{P}} = 0.8$ for the model problem in Section 6.4.1. The values of the last hierarchical estimate η_L refers to Algorithms 5.1v1 and 5.1v2 whereas the values of the last two-level estimate τ_L refers to Algorithms 6.1v1 and 6.1v2.

both sets of marking parameters and versions of the algorithms, on the one hand, the final index sets generated are the same, with $\#\mathcal{P}_L = 25$ indices when running Algorithms 5.1v1 and 6.1v1 ($\theta_X = 0.5$) and $\#\mathcal{P}_L = 34$ indices when running Algorithms 5.1v2 and 6.1v2 ($\theta_X = 0.25$). On the other hand, the advantages of using two-level estimates for the estimation of the energy error is evident. In fact, observe that Algorithm 6.1v1 produced a less refined final triangulation (with nearly twice less number of elements $\#\mathcal{T}_L$), it takes less iterations, and thus overall takes less computational time (about 77% of time saved) than Algorithm 5.1v1 when $\theta_X = 0.5$; similar observations holds for the second versions of the algorithms in case of $\theta_X = 0.25$ and $\theta_{\mathcal{P}} = 0.8$. One of the reason for which Algorithms 6.1v1 and 6.1v2 are faster is certainly due to the computation of the spatial *two-level* contribution to the total estimate τ_ℓ which speeds up the error estimation steps at each iteration of the loop (see Remark 6.1.1).

Figure 6.2 shows the decay of hierarchical estimates η_ℓ (computed by Algorithms 5.1v1 and 5.1v2) and the decay of two-level estimates τ_ℓ (computed by Algorithms 6.1v1 and 6.1v2) for the corresponding case of marking parameters. In addition, we also plot the associated reference energy errors $\|u_{\mathsf{ref}} - u_\ell^{(\eta)}\|_B$ and $\|u_{\mathsf{ref}} - u_\ell^{(\tau)}\|_B$, where $u_\ell^{(\eta)}$ and $u_\ell^{(\tau)}$ denote the SGFEM solutions computed by Algorithms 5.1v1 and 5.1v2 and Algorithms 6.1v1 and 6.1v2, respectively, whereas u_{ref} is the same reference solution used in the numerical experiment of Section 5.4.2. We notice that while both η_ℓ and τ_ℓ decay with the same overall rate in all computations, two-level estimates τ_ℓ underestimate the energy error more than hierarchical estimates η_ℓ do. This can be also observed by looking at Figure 6.3 which shows the computed effectivity indices ξ_ℓ (see (5.43)) for the sequence of SGFEM solutions $u_\ell^{(\eta)}$ and $u_\ell^{(\tau)}$; overall, we see that two-level estimates τ_ℓ tend to

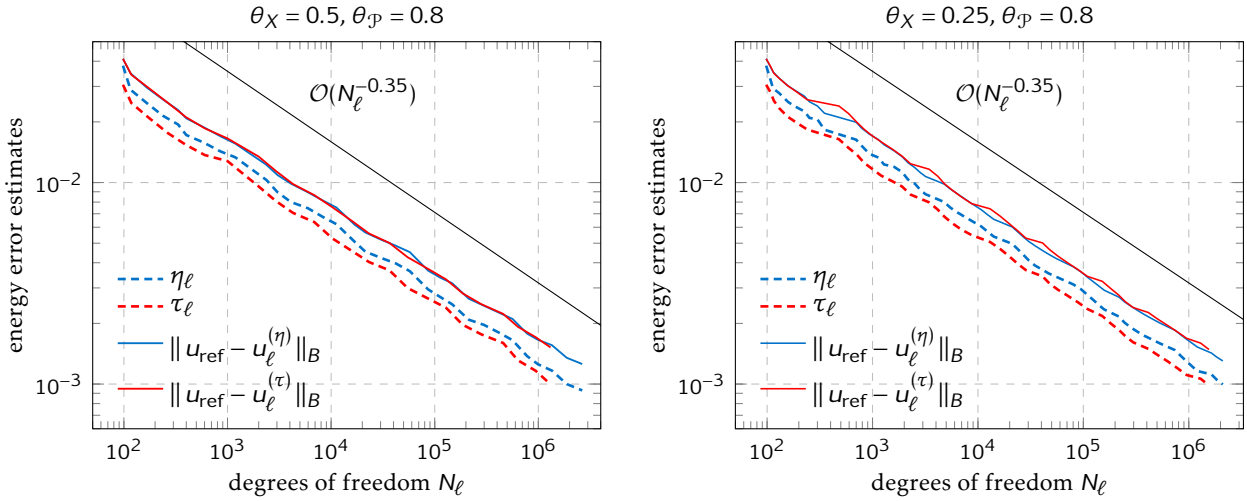


Figure 6.2. Energy error estimates η_ℓ and τ_ℓ, and reference errors $\| u_{\mathrm{ref}} - u_\ell^{(\eta)} \|_B$ and $\| u_{\mathrm{ref}} - u_\ell^{(\tau)} \|_B$ at each step of Algorithms 5.1v1 and 6.1v1 with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.8$ (left) and Algorithms 5.1v2 and 6.1v2 with $\theta_X = 0.25$, $\theta_{\mathcal{P}} = 0.8$ (right), for the model problem in Section 6.4.1. The energy norm of the associated reference solution is $\| u_{\mathrm{ref}} \|_B = 1.90117\text{e-}01$.

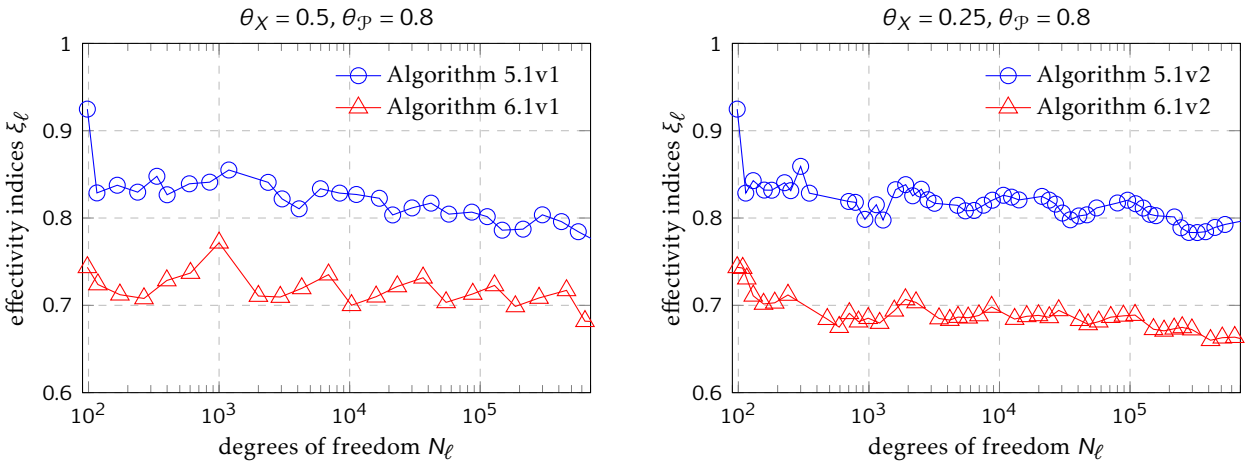


Figure 6.3. The effectivity indices for the SGFEM solutions of the model problem in Section 6.4.1 computed by Algorithms 5.1v1 and 6.1v1 with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.8$ (left) and Algorithms 5.1v2 and 6.1v2 with $\theta_X = 0.25$, $\theta_{\mathcal{P}} = 0.8$ (right).

be close to 0.7 while hierarchical estimates η_ℓ tend to be close to 0.8 as iterations progress (see also Figure 5.6). This is effectively the second reason for which Algorithms 6.1v1 and 6.1v2 achieve the tolerance faster than Algorithms 5.1v1 and 5.1v2 (cf. the final number of degrees of freedom N_L in Table 6.1).

In conclusion, we see that for model problem (4.5), two-level a posteriori error estimates underestimate the true energy error more than hierarchical estimates. On the one hand, running Algorithm 5.1 provides a sequence of overall more accurate error estimates. The cost of this accuracy, however, is borne by the effort of solving extra linear systems for spatial contributions to the

Algorithm 6.1	$\theta_{\mathcal{P}}$								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
v1 ($\theta_X = 0.8$)	2,454,929	2,454,929	2,454,929	2,454,929	2,454,929	2,403,912	1,628,563	**1,560,286**	1,731,044
v2 ($\theta_X = 0.7$)	3,157,697	3,157,697	3,157,697	3,157,697	3,157,697	2,146,095	1,973,460	1,966,801	**1,488,993**
v3 ($\theta_X = 0.7$)	2,094,382	1,891,752	1,970,087	2,014,430	**1,496,851**	1,710,029	1,793,937	2,185,402	1,837,025
v4 ($\theta_X = 0.7$)	2,146,095	1,952,007	2,000,424	1,966,801	**1,460,210***	1,604,638	1,740,662	2,050,900	1,855,200

Table 6.2. Computational cost (6.39) of Algorithms 6.1v1–6.1v4 for the model problem in Section 6.4.2. For each algorithm, we choose the spatial marking parameter $\theta_X \in \Theta$ for which the smallest cost is incurred (see Tables A.1–A.4 in Appendix A) and show the computational cost for all $\theta_{\mathcal{P}} \in \Theta$. The smallest cost for each algorithm is highlighted in boldface in the corresponding row. The boldface starred value is the overall smallest cost, i.e., the smallest cost among all computations with 81 pairs $(\theta_X, \theta_{\mathcal{P}}) \in \Theta \times \Theta$ for all four algorithms.

total estimate at each iteration (see Section 5.3.1). On the other hand, Algorithm 6.1 (in particular, in its versions using Criteria 6.1 and 6.2) is overall more efficient from the computational point of view, as the use of two-level estimates enables the adaptive loop to run faster. At the same time, this still ensures balanced Galerkin approximations and provide competitive results as much as those from Algorithm 5.1.

6.4.2 Experiment 2 - Comparison of computational costs

In this second experiment, we consider the same parametric model problem, posed on the L-shaped domain $D = (-1,1)^2 \setminus (-1,0]^2$, described in Section 5.4.3. We now focus the attention on the performance of adaptive Algorithm 6.1 using the four marking criteria proposed in Section 6.2.4.

We compare Algorithms 6.1v1–6.1v4 with respect to a measure of the total amount of work needed to reach a prescribed tolerance tol. Let $L = L(\mathrm{tol}) \in \mathbb{N}$ denote the last iteration of the algorithm (i.e., such that $\tau_L \leq \mathrm{tol}$) and let N_ℓ be the total number of degrees of freedom at the ℓ-th iteration. We define the computational *cost* of Algorithm 6.1 as the cumulative number of degrees of freedom for all iterations of the adaptive loop, i.e.,

$$\mathrm{cost} = \mathrm{cost}(L) := \sum_{\ell=0}^{L} N_\ell . \tag{6.39}$$

In all computations, we use the initial index set $\mathcal{P}_0 := \{\mathbf{0}\}$, whereas detail index sets are constructed via (5.35) with $M_{\mathcal{Q}} = 1$. In each marking Criteria 6.1–6.4, we set $\vartheta = 1$. Then, we set

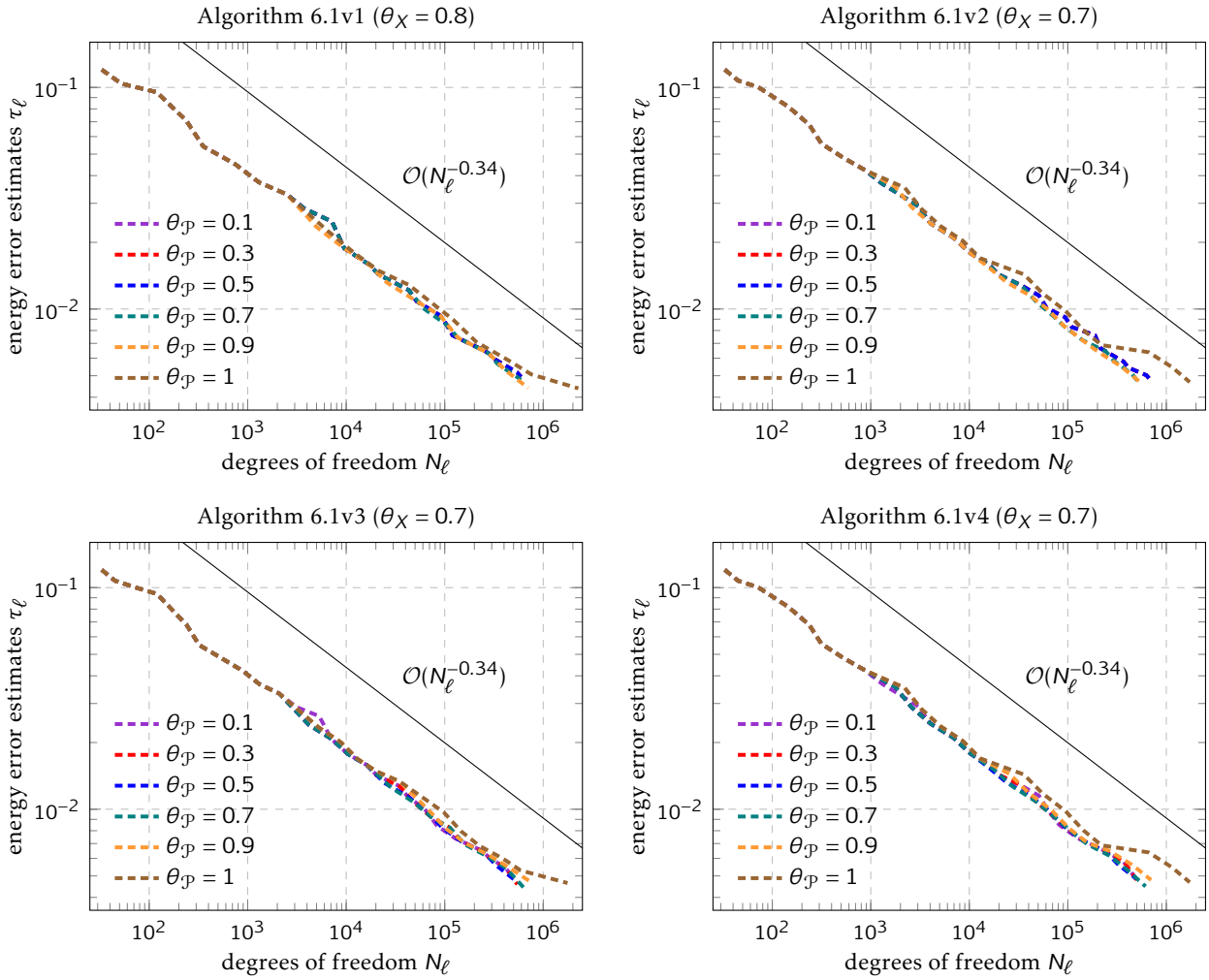


Figure 6.4. Energy error estimates τ_ℓ computed at each iteration of Algorithm 6.1v1 with $\theta_X = 0.8$ and Algorithms 6.1v2–6.1.v4 with $\theta_X = 0.7$, for $\theta_{\mathcal{P}} \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$, for the model problem in Section 6.4.2.

tol = 5e-03 and starting from the coarse triangulation $\mathcal{T}_0$ depicted in Figure 5.7(a), we run Algorithms 6.1v1–6.1v4 with marking parameters $\theta_X, \theta_{\mathcal{P}} \in \Theta := \{0.1, 0.2, \ldots, 0.9\}$; see Figures 5.7(c) and 5.7(d) for a plot of the computed SGFEM mean and variance for this model problem.

The computational costs as well as the empirical convergence rates for each algorithm with 81 pairs $(\theta_X, \theta_{\mathcal{P}}) \in \Theta \times \Theta$ of marking parameters are shown in Tables A.1–A.4 reported in Appendix A. A snapshot of these results is presented in Table 6.2. The results show that the overall smallest cost is achieved by Algorithm 6.1v4 for the values $\theta_X = 0.7$ and $\theta_{\mathcal{P}} = 0.5$. These values of marking parameters are the ones for which also Algorithm 6.1v3 yields the smallest cost among all pairs $(\theta_X, \theta_{\mathcal{P}}) \in \Theta \times \Theta$. This similarity does not hold for Algorithms 6.1v1 and 6.1v2, for which the smallest cost is achieved with $\theta_X = \theta_{\mathcal{P}} = 0.8$ for Algorithm 6.1v1 and with $\theta_X = 0.7$ and $\theta_{\mathcal{P}} = 0.9$ for

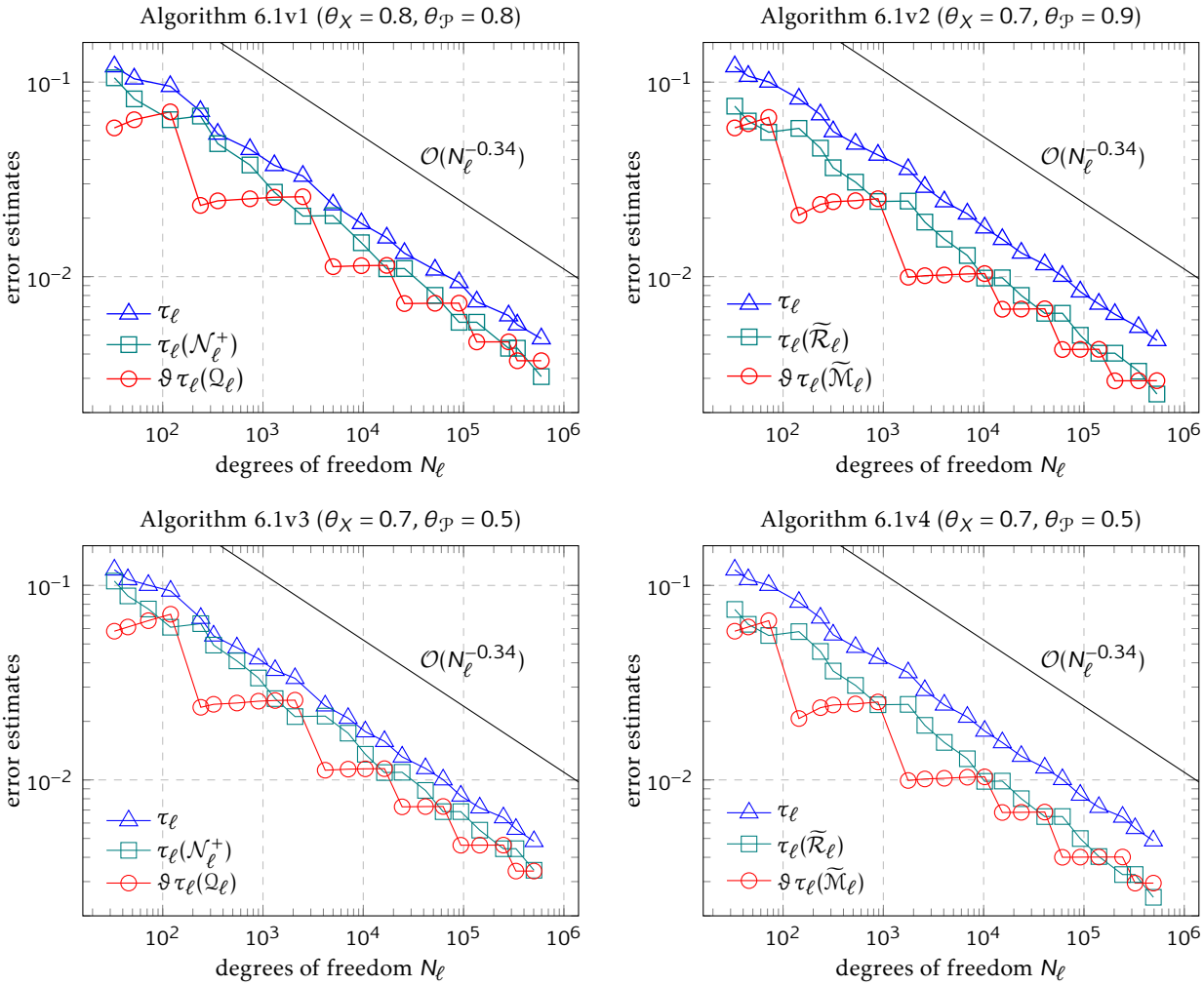


Figure 6.5. Total and local error estimates computed at each iteration of Algorithms 6.1v1–6.1v4 with the marking parameters $\theta_X, \theta_{\mathcal{P}} \in \Theta$ that yield smallest cost (see Table 6.2) for the model problem in Section 6.4.2.

Algorithm 6.1v2. Thus, we conclude that, for the above values of marking parameters, the adaptive algorithms with refinements driven by dominant error reduction estimates (Algorithms 6.1v2 and 6.1v4) incur less computational costs than their counterparts driven by dominant contributing error estimates (Algorithms 6.1v1 and 6.1v3). On the other hand, the algorithms that employ the maximum strategy for parametric refinement (Algorithms 6.1v3 and 6.1v4) incur less computational costs than their counterparts that use Dörfler marking (Algorithms 6.1v1 and 6.1v2). Overall, the smallest computational cost is incurred by the algorithm that combines these two winning strategies, i.e., Algorithm 6.1v4.

Figure 6.4 shows the decay of the overall error estimate τ_ℓ versus the number of degrees of freedom N_ℓ for different values of $\theta_{\mathcal{P}} \in \Theta$, with $\theta_X = 0.8$ in Algorithm 6.1v1 and $\theta_X = 0.7$ in

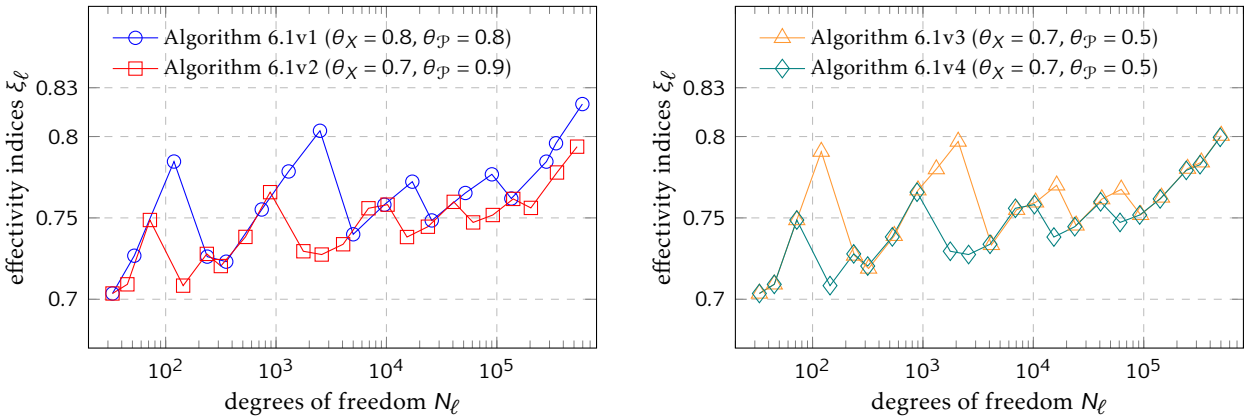


Figure 6.6. The effectivity indices ξ_ℓ for the SGFEM solutions at each iteration of Algorithms 6.1v1 and 6.1v2 (left) and Algorithms 6.1v3 and 6.1v4 (right) with the marking parameters $\theta_X, \theta_{\mathcal{P}} \in \Theta$ that yield smallest cost (see Table 6.2).

Algorithms 6.1v2–6.1v4. The aim of these plots is to show that the sequence of two-level energy error estimates computed by the adaptive algorithm converges regardless of the marking criterion and the value of $\theta_{\mathcal{P}}$ used (see Theorem 6.2); similar decay rates are obtained for other values of $\theta_X, \theta_{\mathcal{P}} \in \Theta$ (see Appendix A). Observe that τ_ℓ decays also in the case $\theta_{\mathcal{P}} = 1 \notin \Theta$ for all algorithms. However, in this case, significantly more degrees of freedom are needed to reach the prescribed tolerance, compared to the cases of $\theta_{\mathcal{P}} \in \Theta$. This is because, for $\theta_{\mathcal{P}} = 1$, each parametric enrichment is performed by augmenting the index set $\mathcal{P}_\ell$ with the whole detail index set $\mathcal{Q}_\ell$.

In Figure 6.5, we plot the decay of all error estimates computed by the four algorithms with the pairs of marking parameters yielding the corresponding smallest cost. As expected, we see that the decay rates of τ_ℓ are similar in all four cases.

To conclude, we test the effectiveness of the error estimation strategy by computing the effectivity indices ξ_ℓ defined in (5.43) (see Section 5.4.1). In particular, we choose $\mathcal{T}_{\mathsf{ref}}$ to be the uniform refinement of the mesh $\mathcal{T}_L$ generated by Algorithm 6.1v2 with $\theta_{\mathcal{P}} = 0.5$ (i.e., one of the final triangulations with the largest number of elements) and $\mathcal{P}_{\mathsf{ref}}$ to be the final index set $\mathcal{P}_L$ produced by Algorithm 6.1v4 with $\theta_{\mathcal{P}} = 0.8$ (i.e., one of the largest index sets generated). Figure 6.6 shows the effectivity indices ξ_ℓ for Algorithms 6.1v1 and 6.1v2 (left) and Algorithms 6.1v3 and 6.1v4 (right) with the pairs of parameters $(\theta_X, \theta_{\mathcal{P}})$ for which the smallest cost is attained. As for hierarchical estimates, and as observed in previous experiment in Section 6.4.1, we see that, in all cases, the error is slightly underestimated as the effectivity indices vary in a range between 0.7 and 0.82 throughout all iterations.

Chapter 7

Adaptive algorithms for goal-oriented error estimation

When adaptive finite element algorithms are used in practical applications, the design of a posteriori error estimation strategies should aim at estimating the error committed in approximating the physical quantity considered useful in the given model. For example, in both Chapters 5 and 6, we focused on the a posteriori estimation of the energy norm of the global error under the SGFEM setting for parametric elliptic problems. Here, associated local error estimates were used to enhance the computed solution and adaptively drive the sequence of energy error estimates to zero. However, in other practical applications, simulations may be oriented to the numerical approximation of a specific (e.g., localised) feature of the solution, which is then often referred to as *quantity of interest*, represented using a linear functional of the solution. In these cases, the energy norm may give very little useful information about the simulation error, thus the error estimation strategy should be designed appropriately.

Error estimation techniques, such as *goal-oriented* error estimations, have been therefore developed for the purpose of controlling the errors in the quantity of interest. For deterministic PDEs, these techniques and the associated adaptive algorithms are very well studied (see, e.g., [21, 109, 22, 74, 11] for the a posteriori error estimation and [95, 20, 81, 67] for a rigorous convergence analysis of adaptive algorithms), whereas less work has been done for PDEs with parametric or uncertain inputs (see, e.g., [90, 37, 3, 57, 36]). In this chapter, our main aim is then to design an adaptive SGFEM algorithm for accurate approximation of moments of a quantity of interest $Q(u)$. Here, u is the solution to parametric model problem (4.5) and Q is a *linear* functional of u. In particular, we are interested in estimating and controlling the expected error in the quantity of interest, i.e., $\mathbb{E}[Q(u-u_{X^{\mathcal{P}}})]$, where $u_{X^{\mathcal{P}}}$ is the SGFEM approximation satisfying the associated weak

problem (4.36). This enables us to use the ideas of goal-oriented adaptivity, where the aim is to control the error in the *goal functional* $G(u) := \mathbb{E}[Q(u(\cdot, \mathbf{y}))]$ rather than in the energy norm.

In what follows, we introduce the goal-oriented error estimation strategy in Section 7.1 and present the goal-oriented adaptive algorithm in Section 7.2. The effectiveness of the error estimation strategy and the performance of the proposed algorithm are tested numerically in Section 7.3 for three representative model problems with parametric coefficients and for three quantities of interest including estimation of directional derivatives and approximation of pointwise values.

7.1 Goal-oriented a posteriori error estimation

We recall the setting of goal-oriented a posteriori error estimation via duality approach (see, e.g., [22, 74]). Firstly, we describe the idea in a pure abstract functional analytic setting of a well-posed problem on a Hilbert space and, secondly, we show how this setting fits the framework of SGFEM approximations for parametric model problem (4.5). The method that we describe will also motivate the design of the goal-oriented adaptive algorithm that is introduced in Section 7.2.

7.1.1 Abstract setting

Let V be a Hilbert space and denote by V' its dual space. Let $B : V \times V \to \mathbb{R}$ be a continuous, symmetric, and coercive bilinear form with associated (energy) norm $\|\cdot\|_B := B(\cdot,\cdot)^{1/2}$. Given two continuous linear functionals $F, G \in V'$, our aim is to approximate $G(u)$, where $u \in V$ is the unique solution to the *primal* problem:

$$B(u, v) = F(v) \qquad \forall\, v \in V. \tag{7.1}$$

To this end, the standard duality approach (see, e.g., [22, 74, 11]) considers $z \in V$ as the unique solution to the *dual* problem:

$$B(v, z) = G(v) \qquad \forall\, v \in V. \tag{7.2}$$

Let $V_\star$ be a finite dimensional subspace of V. Let $u_\star \in V_\star$ and $z_\star \in V_\star$ be the unique Galerkin approximations of the solutions to the discrete primal and dual problem, respectively, i.e.,

$$B(u_\star, v) = F(v) \quad \text{and} \quad B(v, z_\star) = G(v) \qquad \forall\, v \in V_\star.$$

Then, it follows that

$$|G(u)-G(u_\star)| = |B(u-u_\star,z)| = |B(u-u_\star,z-z_\star)| \leq \|u-u_\star\|_B \|z-z_\star\|_B, \tag{7.3}$$

where the second equality holds due to Galerkin orthogonality (cf. (5.2)). Assume that $\mu_\star$ and $\zeta_\star$ are reliable estimates for the energy errors $\|u-u_\star\|_B$ and $\|z-z_\star\|_B$, respectively, i.e.,

$$\|u-u_\star\|_B \lesssim \mu_\star \quad \text{and} \quad \|z-z_\star\|_B \lesssim \zeta_\star. \tag{7.4}$$

Hence, inequality (7.3) implies that the product $\mu_\star\zeta_\star$ is a reliable error estimate for the approximation error in the goal functional:

$$|G(u)-G(u_\star)| \lesssim \mu_\star\zeta_\star. \tag{7.5}$$

7.1.2 Extension to the parametric setting

Let us show how the abstract result on goal-oriented error estimation of Section 7.1.1 can be formulated in the context of SGFEM discretisations for parametric model problem (4.5).

Let $u \in L^2_\pi(\Gamma;H^1_0(D))$ be the unique *primal* solution to primal problem (4.17). Given $g \in H^{-1}(D)$, consider the *quantity of interest*

$$Q(u(\cdot,\mathbf{y})) := \int_D g(\mathbf{x})u(\mathbf{x},\mathbf{y})\,d\mathbf{x} \qquad \forall\, \mathbf{y}\in\Gamma. \tag{7.6}$$

We are interested in approximating the goal functional $G \in L^2_\pi(\Gamma;H^{-1}(D))$ defined as the mean of the quantity of interest, i.e.,

$$G(v) := \mathbb{E}[Q(v(\cdot,\mathbf{y}))] = \int_\Gamma\int_D g(\mathbf{x})v(\mathbf{x},\mathbf{y})\,d\mathbf{x}\,d\pi(\mathbf{y}) \qquad \forall\, v\in L^2_\pi(\Gamma;H^1_0(D)). \tag{7.7}$$

Now, let $z \in L^2_\pi(\Gamma;H^1_0(D))$ be the unique *dual* solution to the dual problem

$$B(v,z) = G(v) \qquad \forall\, v\in L^2_\pi(\Gamma;H^1_0(D)). \tag{7.8}$$

Consider the finite-dimensional subspace $V_{\mathcal{X}\mathcal{P}} \subset V \simeq L^2_\pi(\Gamma;H^1_0(D))$ defined in (4.35) (with V defined in (4.23)) and let $u_{\mathcal{X}\mathcal{P}} \in V_{\mathcal{X}\mathcal{P}}$ be the primal Galerkin approximation satisfying discrete problem (4.36). Then, let $z_{\mathcal{X}\mathcal{P}} \in V_{\mathcal{X}\mathcal{P}}$ be the dual Galerkin approximation satisfying

$$B(v,z_{\mathcal{X}\mathcal{P}}) = G(v) \qquad \forall\, v\in V_{\mathcal{X}\mathcal{P}}. \tag{7.9}$$

Recall that the two-level error estimate $\tau_{\mathcal{X}\mathcal{P}}$ defined in (6.5) provides an efficient and reliable estimate for the energy error in the Galerkin approximation of primal solution u (see Theorem 6.1). Analogously, let $\zeta_{\mathcal{X}\mathcal{P}}$ be the corresponding two-level estimate defined as in (6.5) for the

energy error in the Galerkin approximation of dual solution $z \in V$ (see Remark 7.1.1 below). It follows from (6.7) that

$$\|u - u_{X\mathcal{P}}\|_B \lesssim \tau_{X\mathcal{P}} \qquad \text{and} \qquad \|z - z_{X\mathcal{P}}\|_B \lesssim \zeta_{X\mathcal{P}}. \tag{7.10}$$

From the abstract result in Section 7.1.1 (see (7.3)–(7.5)), we therefore conclude that the error in approximating $G(u)$ can be controlled by the product of the two energy error estimates $\tau_{X\mathcal{P}}$ and $\zeta_{X\mathcal{P}}$, i.e.,

$$|G(u) - G(u_{X\mathcal{P}})| \leq \|u - u_{X\mathcal{P}}\|_B \, \|z - z_{X\mathcal{P}}\|_B \lesssim \tau_{X\mathcal{P}} \zeta_{X\mathcal{P}}. \tag{7.11}$$

Remark 7.1.1. *Note that we define the two-level error estimate $\zeta_{X\mathcal{P}}$ satisfying* (7.10) *exactly as in* (6.5) *since, in our parametric setting, the bilinear form $B(\cdot,\cdot)$ is symmetric (see* (4.18)–(4.20)*). In particular, it follows that the differential operators of the PDEs associated with primal problem* (4.17) *and dual problem* (7.8)*, respectively, are both equal to $-\nabla \cdot (a\nabla v)$ for all $v \in V$. This means that the left-hand side matrices of linear systems arising from discrete problems* (4.36) *and* (7.9) *are the same (see* (4.44)*).*

Remark 7.1.2. *Recall that the hierarchical a posteriori error estimate $\eta_{X\mathcal{P}}$ defined in* (5.22) *is an efficient and reliable estimate for the energy error in the Galerkin approximation of primal solution u (see Proposition 5.1). Analogously, $\eta_{X\mathcal{P}}$ is also an efficient and reliable estimator for the energy error $\|z - z_{X\mathcal{P}}\|_B$ of the associated dual problem (due to the symmetry of the bilinear form $B(\cdot,\cdot)$, see Remark 7.1.1). However, in the reminder of the chapter, we only focus on goal-oriented estimation by means of two-level error estimates.*

7.2 A goal-oriented adaptive algorithm

In this section, we present an adaptive goal-oriented algorithm for the error estimation of goal functional (7.7) for a quantity of interest of the solution to parametric model problem (4.5). In the same spirit of Algorithms 5.1 and 6.1, the goal-oriented algorithm is built under the same framework of adaptive FEM loop (2.13). In particular, at each iteration, the loop consists of (i) two discrete problems (primal and dual) that have to be solved, (ii) two energy error estimations for the computed primal and dual solutions (see (7.10)), and (iii) a marking and a refinement strategy for the control of the error in the approximation of goal functional G defined in (7.7). In what follows, we consider first-order spatial Galerkin approximations.

7.2.1 Local error estimates in the energy norm

Let us introduce the notation for local error estimates in the energy norm for discrete dual problem (7.9).

Similarly to decomposition (6.33) for the energy error estimate $\tau_{X\mathcal{P}}$, we write the two-level error estimate $\zeta_{X\mathcal{P}}$ associated with dual Galerkin solution $z_{X\mathcal{P}} \in V_{X\mathcal{P}}$ satisfying (7.9) as follows. Recall that $\mathcal{N}^+$ denotes the set of midpoints of the interior edges of the underlying triangulation $\mathcal{T}$ of D (see (6.1)) and let $\mathcal{Q}$ be the finite detail index set (5.35). Then,

$$\zeta_{X\mathcal{P}}^2 \;=\; \zeta_{X\mathcal{P}}^2(\mathcal{N}^+,\mathcal{Q}) \;=\; \zeta_{Y\mathcal{P}}(\mathcal{N}^+)^2 \;+\; \zeta_{X\mathcal{Q}}(\mathcal{Q})^2\,, \tag{7.12}$$

with the spatial contribution defined by

$$\zeta_{Y\mathcal{P}}(\mathcal{N}^+)^2 \;:=\; \sum_{\boldsymbol{z}\in\mathcal{N}^+} \zeta_{Y\mathcal{P}}(\boldsymbol{z})^2 \quad \text{with} \quad \zeta_{Y\mathcal{P}}(\boldsymbol{z})^2 \;:=\; \sum_{\nu\in\mathcal{P}} \frac{|\,G(\psi_{\boldsymbol{z}}P_\nu) - B(z_{X\mathcal{P}},\psi_{\boldsymbol{z}}P_\nu)|^2}{\|\,a_0^{1/2}\nabla\psi_{\boldsymbol{z}}\|^2_{L^2(D)}}\,, \tag{7.13}$$

and the parametric contribution defined by

$$\zeta_{X\mathcal{Q}}(\mathcal{Q})^2 \;:=\; \sum_{\mu\in\mathcal{Q}} \zeta_{X\mathcal{Q}}(\mu)^2 \qquad \text{with} \qquad \zeta_{X\mathcal{Q}}(\mu) \;:=\; \|\,e_{X\mathcal{Q}}^{(\mu)}\|_{B_0}\,, \tag{7.14}$$

(cf. (6.31) and (6.32)). Two remarks are needed here. Firstly, notice that in the spatial contribution $\zeta_{Y\mathcal{P}}(\boldsymbol{z})$ defined in (7.13) we write $B(z_{X\mathcal{P}},\psi_{\boldsymbol{z}}P_\nu)$ instead of $B(\psi_{\boldsymbol{z}}P_\nu,z_{X\mathcal{P}})$ due to the symmetry of the bilinear form $B(\cdot,\cdot)$ (see (4.18)–(4.20)). Secondly, we see that the individual parametric estimator $e_{X\mathcal{Q}}^{(\mu)} \in X\otimes\mathcal{P}_\mu$ defined in (7.14), $\mu\in\mathcal{Q}$, satisfies the following problem

$$B_0(e_{X\mathcal{Q}}^{(\mu)},v) \;=\; -B(z_{X\mathcal{P}},v) \qquad \forall\; v\in X\otimes\mathcal{P}_\mu\,. \tag{7.15}$$

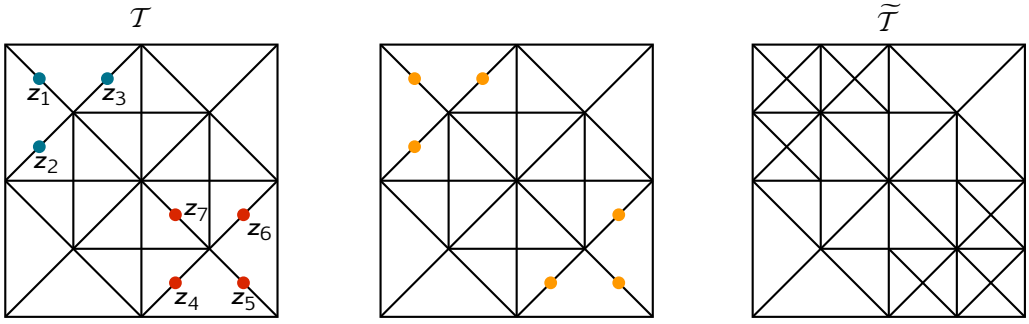
Equation (7.15) is the discrete problem, analogous to (5.27) (see Remark 5.2.1), arising from the residual error equation associated to the dual problem, i.e.,

$$B(z-z_{X\mathcal{P}},v) \;=\; G(v)-B(z_{X\mathcal{P}},v) \qquad \forall\; v\in V\,, \tag{7.16}$$

(cf. (5.1)). Notice that in writing (7.15) and (7.16), we are exploiting again the symmetry of the bilinear form $B(\cdot,\cdot)$.

7.2.2 Marking strategy

In order to compute a more accurate Galerkin solution (and, hence, to reduce the error in the quantity of interest), an enriched approximation space has to be constructed. As for Algorithms 5.1 and 6.1, in the algorithm presented below, the approximation space is enriched at each iteration



$\mathcal{M}^u = \{z_1, z_2, z_3\}$
$\mathcal{M}^z = \{z_4, z_5, z_6, z_7\}$
(a)

$\mathcal{M}_\star := \mathcal{M}^u, \quad \mathcal{M}^\star := \mathcal{M}^z$
$\mathcal{M} := \mathcal{M}_\star \cup \{z_4, z_5, z_6\} = \{\bullet\}$
(b)

(c)

Figure 7.1. Goal-oriented marking strategy on a given triangulation $\mathcal{T}$. (a) Let $\mathcal{M}^u = \{z_1, z_2, z_3\} \subseteq \mathcal{N}^+$ (blue midpoints) and $\mathcal{M}^z = \{z_4, z_5, z_6, z_7\} \subseteq \mathcal{N}^+$ (red midpoints). Suppose that midpoints in $\mathcal{M}^u$ are sorted according to associated estimates $\tau_{Y\mathcal{P}}(z_i)$, $i = 1, 2, 3$, with descendent magnitude, i.e., $\tau_{Y\mathcal{P}}(z_1) \geq \tau_{Y\mathcal{P}}(z_2) \geq \tau_{Y\mathcal{P}}(z_3)$; similarly for midpoints in $\mathcal{M}^z$; (b) Then, $\mathcal{M}_\star := \mathcal{M}^u$ (hence $\mathcal{M}^\star := \mathcal{M}^z$) and $\mathcal{M}$ is given by the union of $\mathcal{M}_\star$ and the $\#\mathcal{M}_\star = 3$ midpoints of $\mathcal{M}^\star$ associated with largest estimates (i.e., z_4, z_5, and z_6); (c) The refined triangulation $\widetilde{\mathcal{T}}$ obtained by the introduction of marked midpoints (orange dots in (b)).

of the adaptive loop either by performing a local refinement of $\mathcal{T}$ or by adding new indices to the index set $\mathcal{P}$. In the former case, the refinement is guided by a set $\mathcal{M} \subseteq \mathcal{N}^+$ of marked midpoints, whereas in the latter case, a set $\mathcal{M} \subseteq \mathcal{Q}$ of marked indices is added to $\mathcal{P}$.

Let us focus on the case of spatial marking of midpoints of $\mathcal{N}^+$. We use the Dörfler marking strategy (see Strategy 2.2) for the two sets $\{\tau_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}^+}$ and $\{\zeta_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}^+}$ of spatial error estimates (see (6.31) and (7.13)) in order to identify two independent sets of marked midpoints associated with primal and dual problem (4.36) and (7.9), respectively. Specifically, given the marking parameter $\theta_X \in (0, 1]$, we build two subsets $\mathcal{M}^u, \mathcal{M}^z \subseteq \mathcal{N}^+$ with minimal cardinality satisfying, respectively,

$$\tau_{Y\mathcal{P}}(\mathcal{M}^u)^2 := \sum_{z\in\mathcal{M}^u} \tau_{Y\mathcal{P}}(z)^2 \geq \theta_X \tau_{Y\mathcal{P}}(\mathcal{N}^+)^2 \quad \text{and} \quad \zeta_{Y\mathcal{P}}(\mathcal{M}^z)^2 := \sum_{z\in\mathcal{M}^z} \zeta_{Y\mathcal{P}}(z)^2 \geq \theta_X \zeta_{Y\mathcal{P}}(\mathcal{N}^+)^2,$$

where $\tau_{Y\mathcal{P}}(\mathcal{N}^+)$ is defined in (6.31) and $\zeta_{Y\mathcal{P}}(\mathcal{N}^+)$ is defined in (7.13). The set $\mathcal{M}^u$ (resp. $\mathcal{M}^z$) satisfying the condition above is returned by a MARK subroutine with θ_X and $\{\tau_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}^+}$ (resp. $\{\zeta_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}^+}$) as inputs (cf. (5.41)). Then, in order to combine the two sets $\mathcal{M}^u$ and $\mathcal{M}^z$, the goal-oriented adaptive algorithm employs the marking strategy adopted in [67]. Comparing the cardinality of $\mathcal{M}^u$ and the cardinality of $\mathcal{M}^z$, we define

$$\begin{aligned} &\mathcal{M}_\star := \mathcal{M}^u \quad \text{and} \quad \mathcal{M}^\star := \mathcal{M}^z \qquad \text{if } \#\mathcal{M}^u \leq \#\mathcal{M}^z, \\ &\mathcal{M}_\star := \mathcal{M}^z \quad \text{and} \quad \mathcal{M}^\star := \mathcal{M}^u \qquad \text{otherwise.} \end{aligned}$$

Marking criterion for goal-oriented adaptive SGFEM

Input: *error estimates* $\{\tau_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}^+}$, $\{\tau_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}}$ *for the primal problem; error estimates* $\{\zeta_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}^+}$, $\{\zeta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}}$ *for the dual problem; marking parameters* $\theta_X, \theta_{\mathcal{P}} \in (0,1]$.
DO

set $\mathcal{M}^u := \mathsf{MARK}\big(\{\tau_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}^+}, \theta_X\big)$ *and* $\mathcal{M}^z := \mathsf{MARK}\big(\{\zeta_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}^+}, \theta_X\big)$;

set $\mathcal{M}^u := \mathsf{MARK}\big(\{\tau_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}}, \theta_{\mathcal{P}}\big)$ *and* $\mathcal{M}^z := \mathsf{MARK}\big(\{\zeta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}}, \theta_{\mathcal{P}}\big)$;

IF $\#\mathcal{M}^u \leq \#\mathcal{M}^z$

set $\mathcal{M}_\star := \mathcal{M}^u$ *and* $\mathcal{M}^\star := \mathcal{M}^z$;

ELSE

set $\mathcal{M}_\star := \mathcal{M}^z$ *and* $\mathcal{M}^\star := \mathcal{M}^u$;

END

IF $\#\mathcal{M}^u \leq \#\mathcal{M}^z$

set $\mathcal{M}_\star := \mathcal{M}^u$ *and* $\mathcal{M}^\star := \mathcal{M}^z$;

ELSE

set $\mathcal{M}_\star := \mathcal{M}^z$ *and* $\mathcal{M}^\star := \mathcal{M}^u$;

END

set $\mathcal{M} := \mathcal{M}_\star \cup \overline{\mathcal{M}}$, *where* $\overline{\mathcal{M}}$ *is the set of* $\#\mathcal{M}_\star$ *midpoints of* $\mathcal{M}^\star$ *with largest error estimates;*

set $\mathcal{M} := \mathcal{M}_\star \cup \overline{\mathcal{M}}$, *where* $\overline{\mathcal{M}}$ *is the set of* $\#\mathcal{M}_\star$ *indices of* $\mathcal{M}^\star$ *with largest error estimates;*

define $\mathcal{R} := \widetilde{\mathcal{N}}^\circ \cap \mathcal{N}^+$, *where* $\widetilde{\mathcal{N}}^\circ$ *is associated with* $\widetilde{\mathcal{T}} := \mathsf{REFINE}(\mathcal{T}, \mathcal{M})$.

END
Output: *subset* $\mathcal{M} \subseteq \mathcal{N}^+$ *of midpoints and subset* $\mathcal{M} \subseteq \mathcal{Q}$ *of indices.*

Criterion 7.1. A marking criterion for a goal-oriented adaptive SGFEM algorithm driven by two-level error estimates.

Then we define a set $\mathcal{M}$ as the union of $\mathcal{M}_\star$ and those $\#\mathcal{M}_\star$ midpoints $z \in \mathcal{M}^\star$ associated with the largest error estimates. The set $\mathcal{M} \subseteq \mathcal{M}_\star \cup \mathcal{M}^\star \subseteq \mathcal{N}^+$ is the set of marked midpoints used to guide local mesh-refinements in the goal-oriented adaptive algorithm. Notice that with this construction there holds $\mathcal{M}_\star \subseteq \mathcal{M}$ and $\#\mathcal{M} \leq C_{\mathsf{mrk}}\#\mathcal{M}_\star$, with $C_{\mathsf{mrk}} = 2$; see Figure 7.1 for an example of this strategy for spatial marking.

Analogously, in order to identify the set $\mathcal{M} \subseteq \mathcal{Q}$ of marked indices to be added to the index set $\mathcal{P}$, we follow the same marking procedure as described above by replacing $\mathcal{M}$, $\mathcal{N}^+$, z, $\tau_{Y\mathcal{P}}(z)$, $\zeta_{Y\mathcal{P}}(z)$, and θ_X with $\mathcal{M}$, $\mathcal{Q}$, μ, $\tau_{X\mathcal{Q}}(\mu)$, $\zeta_{X\mathcal{Q}}(\mu)$, and $\theta_{\mathcal{P}}$, respectively, ($\theta_{\mathcal{P}} \in (0,1]$).

Remark 7.2.1. *There exist several possibilities for 'combining' the sets* $\mathcal{M}^u$ *and* $\mathcal{M}^z$ *into a single set*

that is used for refinement in a goal-oriented adaptive algorithm; see [95, 20, 81, 67]. For example, for goal-oriented adaptivity in the non-parametric setting with marking of finite elements, [67] proves that the strategies of [95, 20, 67] lead to convergence of adaptive goal-oriented algorithms with optimal algebraic rates, while the strategy from [81] might not. In particular, the marking strategy proposed in [67], which is the one described above for marking of midpoints, is a modification of the strategy in [95], and it has been empirically shown that it is is more effective than the original strategy in [95] with respect to the overall computational cost (see (6.39)*).*

The marking strategy described above for the marking of both midpoints and indices is listed in marking Criterion 7.1.

7.2.3 Error reduction in the product of energy norms

Before describing the adaptive loop, let us now describe the idea behind the REFINE module of the goal-oriented adaptive algorithm. The motivation relies on the fact that the algorithm employs the product of energy errors $\|u - u_{X\mathcal{P}}\|_B \|z - z_{X\mathcal{P}}\|_B$ to control the error in approximating $G(u)$ (see (7.11)).

Let $\widetilde{V}_{X\mathcal{P}} \supset V_{X\mathcal{P}}$ be an enrichment of $V_{X\mathcal{P}}$. The enrichment is based on either a triangulation $\widetilde{\mathcal{T}}$ obtained by mesh-refinement of $\mathcal{T}$, i.e., $\widetilde{V}_{X\mathcal{P}} = X(\widetilde{\mathcal{T}}) \otimes \mathcal{P}_{\mathcal{P}}$, or on a larger index set $\widetilde{\mathcal{P}}$, i.e., $\widetilde{V}_{X\mathcal{P}} = X(\mathcal{T}) \otimes \mathcal{P}_{\widetilde{\mathcal{P}}}$. In particular, triangulation $\widetilde{\mathcal{T}}$ is obtained by local NVB refinements whereas $\widetilde{\mathcal{P}}$ is obtained by adding extra indices to $\mathcal{P}$. Let $\widetilde{u}_{X\mathcal{P}}, \widetilde{z}_{X\mathcal{P}} \in \widetilde{V}_{X\mathcal{P}}$ denote the enhanced primal and dual Galerkin solutions, respectively. According to (7.11), one has

$$|G(u) - G(\widetilde{u}_{X\mathcal{P}})| \leq \|u - \widetilde{u}_{X\mathcal{P}}\|_B \|z - \widetilde{z}_{X\mathcal{P}}\|_B .$$

We want to find the error reduction in the product of the energy norms obtained due to the enrichment.

Similarly to (5.13), for enhanced solutions $\widetilde{u}_{X\mathcal{P}}$ and $\widetilde{z}_{X\mathcal{P}}$, there holds

$$\|u - \widetilde{u}_{X\mathcal{P}}\|_B^2 = \|u - u_{X\mathcal{P}}\|_B^2 - \|u_{X\mathcal{P}} - \widetilde{u}_{X\mathcal{P}}\|_B^2 \qquad \text{and} \qquad \|z - \widetilde{z}_{X\mathcal{P}}\|_B^2 = \|z - z_{X\mathcal{P}}\|_B^2 - \|z_{X\mathcal{P}} - \widetilde{z}_{X\mathcal{P}}\|_B^2 ,$$

where the quantities appearing with minus on the right-hand sides of above equalities denote the error reductions (in the energy norm) achieved if solutions $\widetilde{u}_{X\mathcal{P}}$ and $\widetilde{z}_{X\mathcal{P}}$ are going to be computed

for the primal and the dual problem, respectively. Hence,

$$\begin{aligned}\|u-\widetilde{u}_{X\mathcal{P}}\|_B^2\,\|z-\widetilde{z}_{X\mathcal{P}}\|_B^2 \;=\; &\|u-u_{X\mathcal{P}}\|_B^2\,\|z-z_{X\mathcal{P}}\|_B^2\\ &-\Big(\|u-u_{X\mathcal{P}}\|_B^2\,\|z_{X\mathcal{P}}-\widetilde{z}_{X\mathcal{P}}\|_B^2+\|u_{X\mathcal{P}}-\widetilde{u}_{X\mathcal{P}}\|_B^2\,\|z-z_{X\mathcal{P}}\|_B^2\\ &\qquad-\|u_{X\mathcal{P}}-\widetilde{u}_{X\mathcal{P}}\|_B^2\,\|z_{X\mathcal{P}}-\widetilde{z}_{X\mathcal{P}}\|_B^2\Big).\end{aligned}$$

The equality above shows that the quantity in brackets, i.e.,

$$\begin{aligned}\|u-u_{X\mathcal{P}}\|_B^2\,\|z_{X\mathcal{P}}-\widetilde{z}_{X\mathcal{P}}\|_B^2 \;+\; &\|u_{X\mathcal{P}}-\widetilde{u}_{X\mathcal{P}}\|_B^2\,\|z-z_{X\mathcal{P}}\|_B^2\\ &-\|u_{X\mathcal{P}}-\widetilde{u}_{X\mathcal{P}}\|_B^2\,\|z_{X\mathcal{P}}-\widetilde{z}_{X\mathcal{P}}\|_B^2\,,\end{aligned}\tag{7.17}$$

provides the reduction in the product of energy errors that would be achieved due to enrichment $\widetilde{V}_{X\mathcal{P}}$ of the approximation space $V_{X\mathcal{P}}$. Numerical experiments empirically show that the third term contributing to (7.17), i.e., $-\|u_{X\mathcal{P}}-\widetilde{u}_{X\mathcal{P}}\|_B^2\,\|z_{X\mathcal{P}}-\widetilde{z}_{X\mathcal{P}}\|_B^2$, is normally much smaller (in absolute value) compared to the sum of the first two terms in (7.17) and may thus be neglected. Therefore,

$$\|u-u_{X\mathcal{P}}\|_B^2\,\|z_{X\mathcal{P}}-\widetilde{z}_{X\mathcal{P}}\|_B^2 \;+\; \|u_{X\mathcal{P}}-\widetilde{u}_{X\mathcal{P}}\|_B^2\,\|z-z_{X\mathcal{P}}\|_B^2\,,\tag{7.18}$$

provides a good approximation to the true error reduction (7.17).

Now, recall that Theorem 6.1 provides computable estimates of the energy errors (see (6.7)) and of the energy error reductions (see (6.6)). We can use these results to bound each term in (7.18), thus obtaining a computable numerical estimate of the reduction in the product of energy errors. In particular, we define the following two quantities

$$\rho_X^2 \;:=\; \tau_{X\mathcal{P}}^2\left(\sum_{z\in\mathcal{N}^+}\zeta_{Y\mathcal{P}}^2(z)\right)+\zeta_{X\mathcal{P}}^2\left(\sum_{z\in\mathcal{N}^+}\tau_{Y\mathcal{P}}^2(z)\right),\tag{7.19}$$

$$\rho_{\mathcal{P}}^2 \;:=\; \tau_{X\mathcal{P}}^2\left(\sum_{\mu\in\mathcal{M}}\zeta_{X\mathcal{Q}}^2(\mu)\right)+\zeta_{X\mathcal{P}}^2\left(\sum_{\mu\in\mathcal{M}}\tau_{X\mathcal{Q}}^2(\mu)\right).\tag{7.20}$$

For example, suppose that $\widetilde{V}_{X\mathcal{P}}$ is obtained by a mesh-refinement of $V_{X\mathcal{P}}$ (i.e., by the introduction in $\mathcal{T}$ of all refined midpoints $\mathcal{R}\subseteq\mathcal{N}^+$ by local NVB refinements), then

$$\rho_X^2 \;\simeq\; \|u-u_{X\mathcal{P}}\|_B^2\,\|z_{X\mathcal{P}}-\widetilde{z}_{X\mathcal{P}}\|_B^2 \;+\; \|u_{X\mathcal{P}}-\widetilde{u}_{X\mathcal{P}}\|_B^2\,\|z-z_{X\mathcal{P}}\|_B^2\,.$$

Likewise, the reduction (7.18) due to polynomial enrichment (i.e., by adding the set $\mathcal{M}$ of marked indices to $\mathcal{P}$) is estimated by $\rho_{\mathcal{P}}^2$ defined in (7.20). Thus, by comparing these two estimates (ρ_X and $\rho_{\mathcal{P}}$), the adaptive algorithm chooses the enrichment of $V_{X\mathcal{P}}$ (either mesh-refinement or polynomial enrichment) that corresponds to the larger estimate of the associated error reduction.

7.2.4 Goal-oriented adaptive loop

Let us now briefly describe the goal-oriented adaptive algorithm for numerical approximation of the goal functional $G(u)$ defined in (7.7).

Let $\ell \in \mathbb{N}_0$ denote the iteration counter of the loop. We use ℓ to denote triangulations, index sets, Galerkin solutions, etc., associated with the ℓ-th iteration of the adaptive loop. In particular, $V_\ell := X_\ell \otimes \mathcal{P}_{\mathcal{P}_\ell}$ denotes the finite-dimensional subspace of V, $u_\ell \in V_\ell$ and $\zeta_\ell \in V_\ell$ are the primal and dual Galerkin solutions to (4.36) and (7.9), respectively, and $\tau_\ell := \tau_{X\mathcal{P}}^{(\ell)}$ and $\zeta_\ell := \zeta_{X\mathcal{P}}^{(\ell)}$ are the associated total two-level error estimates (see (6.33) and (7.12)).

For each iteration of the loop, discrete primal problem (4.36) and dual problem (7.9) are solved by the SOLVE subroutine,

$$u_\ell = \mathsf{SOLVE}\,(\mathcal{T}_\ell, \mathcal{P}_\ell, a, f) \quad \text{and} \quad z_\ell = \mathsf{SOLVE}\,(\mathcal{T}_\ell, \mathcal{P}_\ell, a, g)\,,$$

where f and g are the right-hand side data of the primal and the dual problem, respectively. Local two-level spatial and parametric estimates of the error for the two problems are computed by the ESTIMATE subroutine

$$\begin{aligned}\Big[\{\tau_{\mathcal{Y}\mathcal{P}}(z)\}_{z\in\mathcal{N}_\ell^+},\, \{\tau_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}\Big] &= \mathsf{ESTIMATE}\Big(u_\ell, \mathcal{T}_\ell, \mathcal{P}_\ell, \mathcal{Q}_\ell, a, f\Big),\\ \Big[\{\zeta_{\mathcal{Y}\mathcal{P}}(z)\}_{z\in\mathcal{N}_\ell^+},\, \{\zeta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}\Big] &= \mathsf{ESTIMATE}\Big(z_\ell, \mathcal{T}_\ell, \mathcal{P}_\ell, \mathcal{Q}_\ell, a, g\Big),\end{aligned}$$

where the detail index set $\mathcal{Q}_\ell \subset \mathcal{I} \setminus \mathcal{P}_\ell$ is constructed via (5.35). Total two-level estimates τ_ℓ and ζ_ℓ for the primal and the dual problem are assembled using (6.33) and (7.12), respectively, and the product $\tau_\ell\zeta_\ell$ for the goal-oriented estimation of the error $|G(u) - G(u_\ell)|$ is thus obtained (see (7.11)). If a prescribed tolerance tol is met, i.e., if $\tau_\ell\zeta_\ell \le \mathrm{tol}$, then the adaptive process stops. Otherwise, the algorithm employs marking Criterion 7.1 to select a subset $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ of marked midpoints and a subset $\mathcal{M}_\ell \subseteq \mathcal{Q}$ of marked indices. The type of enrichment to pursue is decided by comparing the error reduction estimates $\rho_{X,\ell}$ and $\rho_{\mathcal{P},\ell}$ defined in (7.19) and (7.20) for mesh-refinements and parametric enrichments, respectively. If $\rho_{X,\ell}$ is dominant (i.e., $\rho_{X,\ell} \ge \rho_{\mathcal{P},\ell}$) a larger error reduction (in the product of the energy norms, see Section 7.2.3) is expected, and a new triangulation is obtained via $\mathcal{T}_{\ell+1} := \mathsf{REFINE}(\mathcal{T}_\ell, \mathcal{M}_\ell)$, with the REFINE subroutine implementing local NVB refinements where reference edges are the longest edges of each element; also, the same index set is used on the next iteration, i.e., $\mathcal{P}_{\ell+1} = \mathcal{P}_\ell$. If $\rho_{\mathcal{P},\ell}$ is dominant, then the algorithm uses the same triangulation, i.e., $\mathcal{T}_{\ell+1} = \mathcal{T}_\ell$, and the index set $\mathcal{P}_\ell$ is enriched by adding the set of

Goal-oriented adaptive SGFEM algorithm

Input: *data* a, f, g; *triangulation* $\mathcal{T}_0$, *index set* $\mathcal{P}_0$; *marking parameters* θ_X, $\theta_{\mathcal{P}}$; *tolerance* tol.
FOR $\ell = 0, 1, 2, \dots$ DO

$u_\ell = \mathsf{SOLVE}(\mathcal{T}_\ell, \mathcal{P}_\ell, a, f)$;

$z_\ell = \mathsf{SOLVE}(\mathcal{T}_\ell, \mathcal{P}_\ell, a, g)$;

$\Big[\{\tau_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}_\ell^+},\ \{\tau_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}\Big] = \mathsf{ESTIMATE}\Big(u_\ell, \mathcal{T}_\ell, \mathcal{P}_\ell, \mathcal{Q}_\ell, a, f\Big)$;

$\Big[\{\zeta_{Y\mathcal{P}}(z)\}_{z\in\mathcal{N}_\ell^+},\ \{\zeta_{X\mathcal{Q}}(\mu)\}_{\mu\in\mathcal{Q}_\ell}\Big] = \mathsf{ESTIMATE}\Big(z_\ell, \mathcal{T}_\ell, \mathcal{P}_\ell, \mathcal{Q}_\ell, a, g\Big)$;

$\tau_\ell = \Big(\tau_{Y\mathcal{P}}(\mathcal{T}_\ell)^2 + \tau_{X\mathcal{Q}}(\mathcal{Q}_\ell)^2\Big)^{1/2}$;

$\zeta_\ell = \Big(\zeta_{Y\mathcal{P}}(\mathcal{T}_\ell)^2 + \zeta_{X\mathcal{Q}}(\mathcal{Q}_\ell)^2\Big)^{1/2}$;

IF $\tau_\ell\zeta_\ell \le \mathrm{tol}$ THEN BREAK; END

obtain $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell^+$ *and* $\mathcal{M}_\ell \subseteq \mathcal{Q}_\ell$ *by using Criterion 7.1;*

compute the error reduction estimates $\rho_{X,\ell}$ *and* $\rho_{\mathcal{P},\ell}$ *(see* (7.19) *and* (7.20)*);*

IF $\rho_{X,\ell} \ge \rho_{\mathcal{P},\ell}$

set $\mathcal{T}_{\ell+1} := \mathsf{REFINE}(\mathcal{T}_\ell, \mathcal{M}_\ell)$ *and* $\mathcal{P}_{\ell+1} := \mathcal{P}_\ell$;

ELSE

set $\mathcal{T}_{\ell+1} := \mathcal{T}_\ell$ *and* $\mathcal{P}_{\ell+1} := \mathcal{P}_\ell \cup \mathcal{M}_\ell$.

END

END
Output: *sequence* (u_ℓ, ζ_ℓ) *of primal and dual Galerkin solutions and estimates* $\tau_\ell\zeta_\ell$ *of the error in approximating* $G(u)$.

Algorithm 7.1. Goal-oriented adaptive SGFEM algorithm driven by the products of two-level error estimates in the energy norm.

marked indices $\mathcal{M}_\ell$, i.e., $\mathcal{P}_{\ell+1} = \mathcal{P}_\ell \cup \mathcal{M}_\ell$.

The goal-oriented adaptive algorithm described above is listed in Algorithm 7.1. Note that similarly to Algorithms 5.1 and 6.1, Algorithm 7.1 returns a sequence of adaptively refined triangulations $(\mathcal{T}_\ell)_{\ell\in\mathbb{N}_0}$ associated with nested finite element spaces $(X_\ell)_{\ell\in\mathbb{N}_0}$ as well as a sequence of adaptively enriched nested index sets $(\mathcal{P}_\ell)_{\ell\in\mathbb{N}_0}$.

7.3 Numerical experiments

In this section, we report the results of some numerical experiments that demonstrate the performance of the goal-oriented adaptive Algorithm 7.1 for parametric model problem (4.5). All exper-

iments were performed using the toolbox Stochastic T-IFISS [28] (see Appendix B) on a desktop computer equipped with an Intel Core CPU i5-4590@3.30GHz and 8.00GB of RAM.

7.3.1 Setup of the experiments

We use the representations of f and g as introduced in [95] to define the corresponding functionals $F(v)$ and $G(v)$ in discrete primal and dual problems, respectively (see (4.36) and (7.9)); see also [67, Section 4]. Specifically, let $f_i, g_i \in L^2(D)$, $i = 0,1,2$, and set $\boldsymbol{f} := (f_1, f_2)$ and $\boldsymbol{g} := (g_1, g_2)$. Define

$$F(v) = \int_\Gamma \int_D f_0(\mathbf{x}) v(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\pi(\mathbf{y}) \; - \; \int_\Gamma \int_D \boldsymbol{f}(\mathbf{x}) \cdot \nabla v(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\pi(\mathbf{y}) \qquad \forall\ v \in V, \tag{7.21}$$

and

$$G(v) = \int_\Gamma \int_D g_0(\mathbf{x}) v(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\pi(\mathbf{y}) \; - \; \int_\Gamma \int_D \boldsymbol{g}(\mathbf{x}) \cdot \nabla v(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\pi(\mathbf{y}) \qquad \forall\ v \in V. \tag{7.22}$$

That is, for primal problem (4.36), representation (7.21) arise from considering right-hand side sources of the form $f := f_0 + \nabla \cdot \boldsymbol{f}$, where $f_0 \in H^{-1}(D)$ and $\nabla\cdot : L^2(D) \times L^2(D) \to H^{-1}(D)$ is the divergence operator with weak derivatives. Similarly, for dual problem (7.9), we write $g := g_0 + \nabla \cdot \boldsymbol{g}$; see [95]. The motivation behind these representations is to introduce different non-geometric singularities in the primal and dual solutions. In the context of goal-oriented adaptivity, this emphasises the need for separate marking to resolve singularities in both solutions in different regions of the computational domain.

We run Algorithm 7.1 with initial index set (5.42) with only one active parameter. Detail index sets $\mathcal{Q}_\ell$ are computed via (5.35). Let $L = L(\mathrm{tol}) \in \mathbb{N}$ be the smallest integer such that $\tau_L \zeta_L \leq \mathrm{tol}$, with tol denoting a stopping tolerance. We will collect the same output data as listed in Section 5.4.1 as well as the computational cost defined in (6.39). In order to test the effectiveness of the goal-oriented error estimation, we compare the products $\tau_\ell \zeta_\ell$, $\ell = 0,\dots,L$, with a reference error $|G(u_{\mathrm{ref}}) - G(u_\ell)|$, where $u_{\mathrm{ref}} \in V_{\mathrm{ref}} := X_{\mathrm{ref}} \otimes \mathcal{P}_{\mathcal{P}_{\mathrm{ref}}}$ is an accurate primal solution. As in Section 5.4.1, we compute u_{ref} by employing quadratic (second-order) finite element approximations over a fine triangulation $\mathcal{T}_{\mathrm{ref}}$ (i.e., $X_{\mathrm{ref}} := \mathcal{S}_0^2(\mathcal{T}_{\mathrm{ref}})$) and using a large index set $\mathcal{P}_{\mathrm{ref}}$ (both $\mathcal{T}_{\mathrm{ref}}$ and $\mathcal{P}_{\mathrm{ref}}$ are to be specified in each experiment). Then, the effectivity indices for the goal-oriented estimation are defined by:

$$\varsigma_\ell := \frac{\tau_\ell \zeta_\ell}{|G(u_{\mathrm{ref}}) - G(u_\ell)|}, \qquad \ell = 0,\dots,L. \tag{7.23}$$

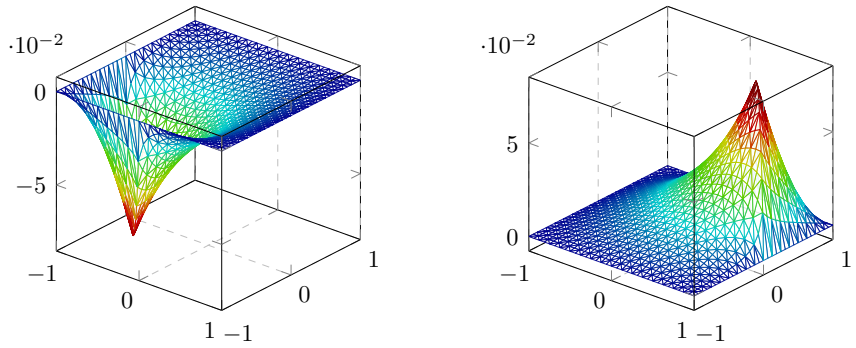


Figure 7.2. The mean fields of primal (left) and dual (right) Galerkin solutions for the model problem in Section 7.3.2.

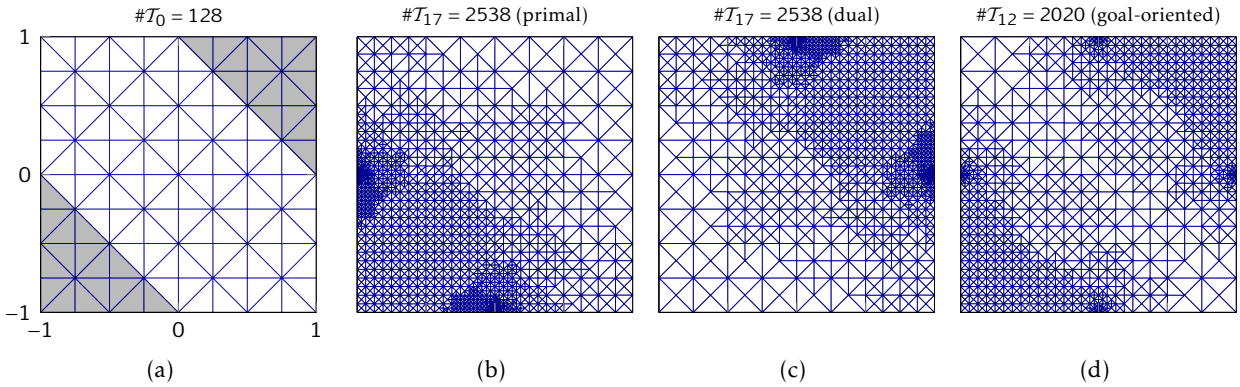


Figure 7.3. Numerical experiment for the model problem in Section 7.3.2. (a) Initial triangulation $\mathcal{T}_0$ with shaded triangles $\mathcal{T}_f$ and $\mathcal{T}_g$; (b)-(c) Triangulations generated by a standard adaptive SGFEM algorithm with spatial refinements driven by either error estimates τ_ℓ or by error estimates ζ_ℓ; (d) Triangulation generated by the goal-oriented adaptive Algorithm 7.1.

7.3.2 Experiment 1 - Estimation of directional derivatives on square domain

In the first experiment, we consider the parametric model problem (4.5) posed on the square domain $D = (-1,1)^2$. Suppose that coefficient $a(\mathbf{x},\mathbf{y})$ in (4.8) is a second-order stationary random field with prescribed (constant) mean $\mathbb{E}[a]$ and covariance $\mathrm{Cov}[a]$ (see Example 3.2.1). In particular, we assume that $\mathrm{Cov}[a]$ is the separable exponential covariance function given by

$$\mathrm{Cov}[a](\mathbf{x},\mathbf{x}') = \sigma^2 \exp\left(-\frac{|x_1 - x_1'|}{\ell_1} - \frac{|x_2 - x_2'|}{\ell_2}\right),$$

where $\mathbf{x} = (x_1, x_2), \mathbf{x}' = (x_1', x_2') \in D$, σ denotes the standard deviation of the random field, and $\ell_1, \ell_2 > 0$ are the correlation lengths (cf. (3.9)). Then we consider a Karhunen–Lòeve expansion of the random field (see Section 3.3.1),

$$a(\mathbf{x},\mathbf{y}) = \mathbb{E}[a](\mathbf{x}) + c\sigma \sum_{m=1}^{\infty} \sqrt{\lambda_m}\varphi_m(x) y_m, \qquad x \in D,\ \mathbf{y} \in \Gamma, \tag{7.24}$$

	$\theta_X = 0.5,\ \theta_{\mathcal{P}} = 0.9$		$\theta_X = 0.25,\ \theta_{\mathcal{P}} = 0.9$	
	$M_{\mathcal{Q}} = 1$	$M_{\mathcal{Q}} = 2$	$M_{\mathcal{Q}} = 1$	$M_{\mathcal{Q}} = 2$
L	20	19	36	34
t (sec)	190	250	309	343
cost	1,584,981	2,398,452	3,073,941	3,798,893
$\tau_L \zeta_L$	6.9126e-06	5.3746e-06	6.0296e-06	6.3484e-06
N_L	368,270	920,873	587,554	797,490
$\#\mathcal{T}_L$	53,184	97,752	54,000	53,748
$\#\mathcal{P}_L$	14	19	22	30
$M_{\mathcal{P}_L}$	8	11	10	15

Table 7.1. The outputs obtained by running Algorithm 7.1 with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$ (case (i)) and $\theta_X = 0.25$, $\theta_{\mathcal{P}} = 0.9$ (case (ii)) for the model problem in Section 7.3.2.

where $\{(\lambda_m, \varphi_m)\}_{m=1}^{\infty}$ are the eigenpairs of the associated covariance operator $\mathcal{C}_a$ (see (3.11)), y_m are the images of pairwise uncorrelated mean-zero random variables, and the constant $c > 0$ is chosen such that $\mathrm{Var}(c\, y_m) = 1$ for all $m \in \mathbb{N}$. Recall that analytical expressions for λ_m and φ_m exist in the one-dimensional case and, as a consequence, the formulas for rectangular domains follow by tensorisation (see Example 3.3.1). In this experiment, we assume that y_m are the images of independent mean-zero random variables on $\Gamma_m = [-1, 1]$ with the following density

$$\rho(y_m) \;=\; (2\Phi(1) - 1)^{-1} \left(\frac{1}{\sqrt{2\pi}} \right) \exp\left(-\frac{y_m^2}{2} \right) \qquad \forall\ m \in \mathbb{N},$$

i.e., the 'truncated' Gaussian density (4.27) (with constants $b = s = 1$) such that the corresponding polynomials which are orthonormal with respect to inner product $(\cdot,\cdot)_{\pi_m}$ ($m \in \mathbb{N}$) are the Rys polynomials (see Example 4.2.3). In this case, we have $c \approx 1.8534$ in (7.24).

We test the performance of Algorithm 7.1 by considering a parametric version of Example 7.3 in [95]. Specifically, let $f_0 = g_0 = 0$, $\boldsymbol{f} = (\chi_{T_f}, 0)$, and $\boldsymbol{g} = (\chi_{T_g}, 0)$, where χ_{T_f} and χ_{T_g} denote the characteristic functions of the following triangles

$$T_f \;:=\; \mathrm{conv}\Big(\{(-1,-1), (0,-1), (-1,0)\} \Big) \qquad \text{and} \qquad T_g \;:=\; \mathrm{conv}\Big(\{(1,1), (0,1), (1,0)\} \Big),$$

respectively (see Figure 7.3(a)). Then, the functionals F and G in (7.21) and (7.22) read as

$$F(v) = -\int_\Gamma \int_{T_f} \frac{\partial v}{\partial x_1}(\mathbf{x}, \mathbf{y})\, d\mathbf{x}\, d\pi(\mathbf{y}) \qquad \text{and} \qquad G(v) = -\int_\Gamma \int_{T_g} \frac{\partial v}{\partial x_1}(\mathbf{x}, \mathbf{y})\, d\mathbf{x}\, d\pi(\mathbf{y}) \qquad \forall\ v \in V.$$

Setting $\sigma = 0.15$, $\ell_1 = \ell_2 = 2.0$, and $\mathbb{E}[a](\mathbf{x}) = 2$ for all $\mathbf{x} \in D$, we compare the performance of Algorithm 7.1 for different input values of marking parameter θ_X as well as parameter $M_{\mathcal{Q}}$ in detail index set (5.35). More precisely, we consider two sets of marking parameters: (i) $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$; (ii) $\theta_X = 0.25$, $\theta_{\mathcal{P}} = 0.9$; in each case, we run Algorithm 7.1 with $M_{\mathcal{Q}} = 1$ and $M_{\mathcal{Q}} = 2$

	$\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$			
	$M_{\mathcal{Q}} = 1$		$M_{\mathcal{Q}} = 2$	
$\mathcal{P}_\ell$	$\ell = 10$	(0 1)	$\ell = 8$	(0 0 1) (0 1 0)
	$\ell = 11$	(0 0 1)	$\ell = 12$	(0 0 0 0 1) (0 0 0 1 0)
	$\ell = 14$	(0 0 0 1) (1 1 0 0) (2 0 0 0)	$\ell = 14$	(0 0 0 0 0 0 1) (0 0 0 0 0 1 0) (1 1 0 0 0 0 0) (2 0 0 0 0 0 0)
	$\ell = 16$	(0 0 0 0 1)	$\ell = 16$	(0 0 0 0 0 0 0 0 1) (0 0 0 0 0 0 0 1 0) (1 0 1 0 0 0 0 0 0)
	$\ell = 18$	(0 0 0 0 0 1) (1 0 1 0 0 0)	$\ell = 18$	(0 0 0 0 0 0 0 0 0 0 1) (0 0 0 0 0 0 0 0 0 1 0) (0 1 1 0 0 0 0 0 0 0 0) (1 0 0 0 0 1 0 0 0 0 0) (1 0 0 0 1 0 0 0 0 0 0) (1 0 0 1 0 0 0 0 0 0 0)
	$\ell = 19$	(0 0 0 0 0 0 1) (1 0 0 1 0 0 0)		
	$\ell = 20$	(0 0 0 0 0 0 0 1) (1 0 0 0 1 0 0 0)		

Table 7.2. The parametric enrichments of the index set $\mathcal{P}$ obtained by running Algorithm 7.1 with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$ (case (i)) for the model problem in Section 7.3.2.

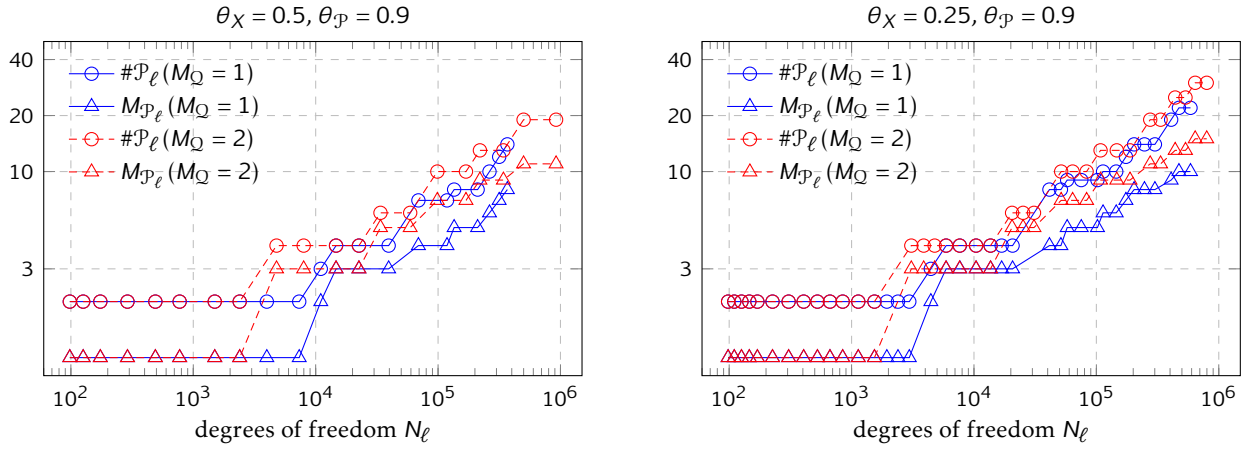


Figure 7.4. Characteristics of the index sets $\mathcal{P}_\ell$ at each iteration of Algorithm 7.1 for the model problem in Section 7.3.2.

in (5.35). The same stopping tolerance is set to tol = 7e-6 in all four computations.

Figure 7.2 (left) shows the mean field of the primal Galerkin solution exhibiting a singularity along the line connecting the points $(-1, 0)$ and $(0, -1)$. Similarly, the mean field of the dual Galerkin solution in Figure 7.2 (right) exhibits a singularity along the line connecting the points $(1, 0)$ and $(0, 1)$.

Figure 7.3(a) shows the initial triangulation $\mathcal{T}_0$ used in this experiment. Figure 7.3(b) and 7.3(c)

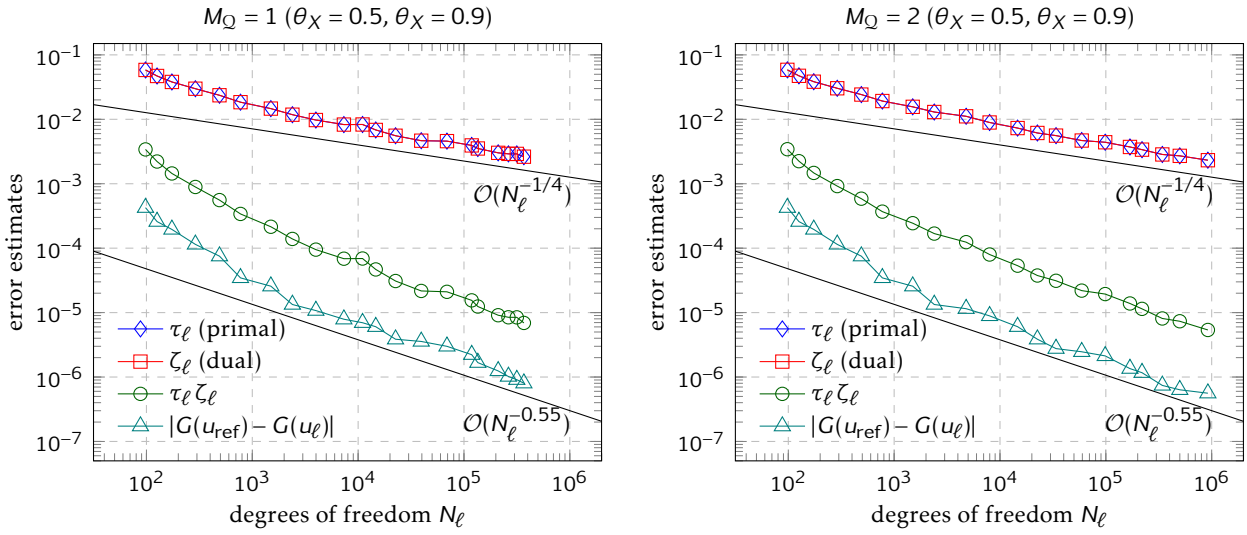


Figure 7.5. Error estimates τ_ℓ, ζ_ℓ, $\tau_\ell\zeta_\ell$ and the reference error $|G(u_{\mathrm{ref}}) - G(u_\ell)|$ at each iteration of Algorithm 7.1 with $\theta_X = 0.5$, $\theta_{\mathcal{P}} = 0.9$ (case (i)) for the model problem in Section 7.3.2. Here, $G(u_{\mathrm{ref}}) = -3.180377\text{e-}03$.

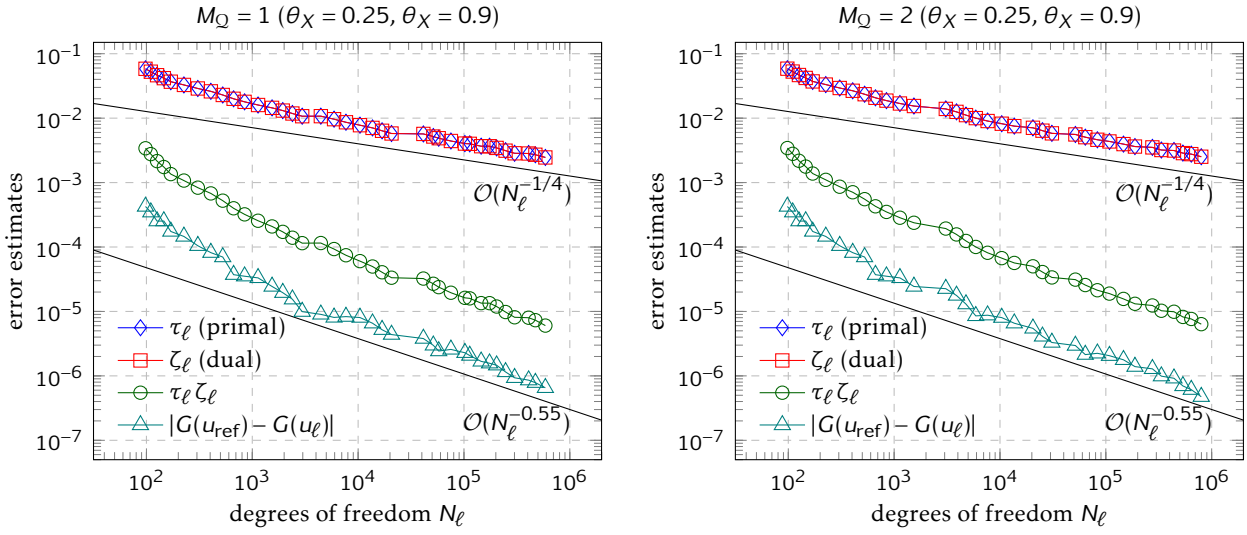


Figure 7.6. Error estimates τ_ℓ, ζ_ℓ, $\tau_\ell\zeta_\ell$ and the reference error $|G(u_{\mathrm{ref}}) - G(u_\ell)|$ at each iteration of Algorithm 7.1 with $\theta_X = 0.25$, $\theta_{\mathcal{P}} = 0.9$ (case (ii)) for the model problem in Section 7.3.2. Here, $G(u_{\mathrm{ref}}) = -3.180377\text{e-}03$.

depict the refined triangulations generated by an adaptive SGFEM algorithm with spatial refinements driven either solely by the estimates τ_ℓ for the error in the primal Galerkin solution or solely by the estimates ζ_ℓ for the error in the dual Galerkin solution. Figure 7.3(d) shows the triangulation produced by Algorithm 7.1. As expected, this triangulation simultaneously captures spatial features of both primal and dual solutions.

In Table 7.1, we collect the final outputs of computations in cases (i) and (ii) for $M_{\mathcal{Q}} = 1$ and $M_{\mathcal{Q}} = 2$, whereas Table 7.2 shows the index set enrichments in case (i). Recall that choosing larger

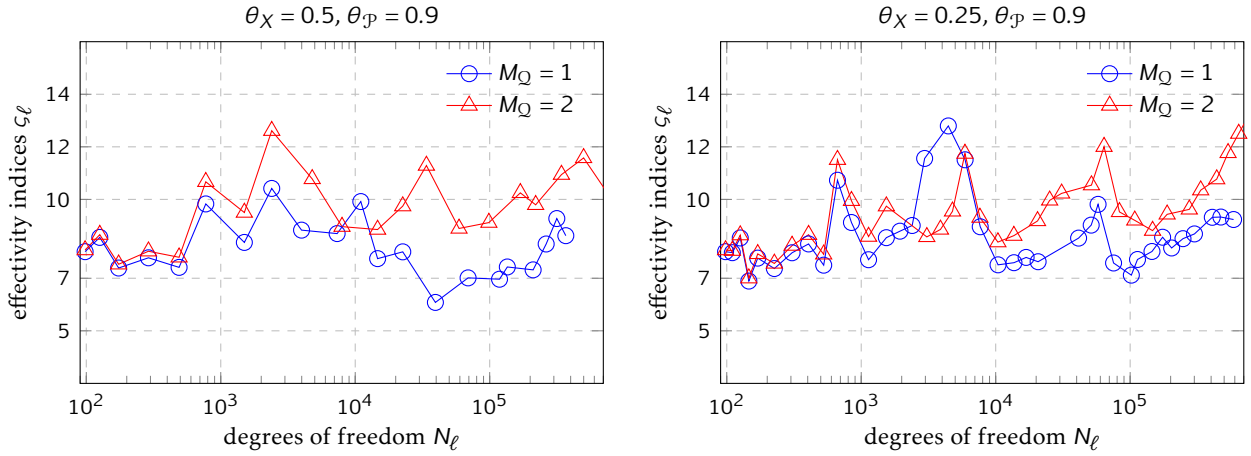


Figure 7.7. The effectivity indices for the goal-oriented error estimates $\tau_\ell \zeta_\ell$ at each iteration of Algorithm 7.1 for the model problem in Section 7.3.2.

values of $M_{\mathcal{Q}}$ in (5.35) leads to larger detail index sets, and thus, larger sets of marked indices, at each iteration. As a result, for $M_{\mathcal{Q}} = 2$, more random variables are active in the final index set and the total number of iterations is reduced (compare the values of L, $\#\mathcal{P}_L$, and $M_{\mathcal{P}_L}$ in Table 7.1). This can be also observed by looking at Figure 7.4 that visualises the evolution of the index set in cases (i) and (ii). Notice that, however, larger detail index sets yield larger computational times due to more expensive computations of the parametric estimates (cf. the times in Table 7.1).

Figure 7.5 shows the convergence history of three error estimates (τ_ℓ, ζ_ℓ, and $\tau_\ell \zeta_\ell$) and the reference error $|G(u_{\text{ref}}) - G(u_\ell)|$ in case (i) for both $M_{\mathcal{Q}} = 1$ and $M_{\mathcal{Q}} = 2$ (see the end of this subsection for details on how the reference solution u_{ref} is computed). We observe that the estimates of the error in approximating $G(u)$ (i.e., the products $\tau_\ell \zeta_\ell$) decay with an overall rate of about $\mathcal{O}(N^{-0.55})$ for both $M_{\mathcal{Q}} = 1$ and $M_{\mathcal{Q}} = 2$. We notice that choosing $M_{\mathcal{Q}} = 2$ has a 'smoothing' effect on the decay of $\tau_\ell \zeta_\ell$ (see Figure 7.5 (right)); this is due to larger index set enrichments in this case compared to those in the case of $M_{\mathcal{Q}} = 1$ (see the evolution of $\mathcal{P}_\ell$ in Table 7.2).

Analogously to Figure 7.5, in Figure 7.6, we plot three error estimates as well as the reference error in the goal functional in case (ii). We observe that $\tau_\ell \zeta_\ell$ decay with about the same overall rate as in case (i), i.e., $\mathcal{O}(N^{-0.55})$. On the other hand, the 'smoothing' effect due to a larger $M_{\mathcal{Q}}$ is less evident in case (ii), compared to case (i). This is likely due to a smaller value of the (spatial) marking parameter θ_X in case (ii), which provides a more balanced refinement of spatial and parametric components of the generated Galerkin approximations (the marking parameter $\theta_{\mathcal{P}}$ is the same in both cases). Notice that in both Figures 7.5 and 7.6 the error estimates τ_ℓ and ζ_ℓ

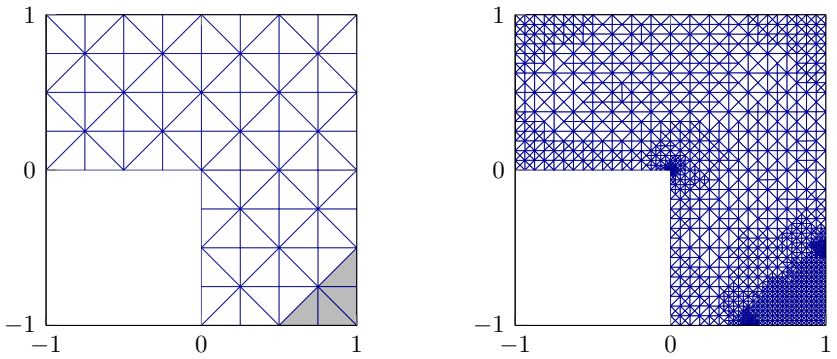


Figure 7.8. Initial triangulation $\mathcal{T}_0$ with shaded triangle T_g (left) and the triangulation generated by Algorithm 7.1 for an intermediate tolerance (right) for the model problem in Section 7.3.3.

coincides due to the symmetry of the problem (cf. the primal and dual solutions in Figure 7.2).

Finally, for all cases considered in this experiment, we compute the effectivity indices (7.23) as explained in Section 7.3.1. Here, the solution u_{ref} is computed using the triangulation $\mathcal{T}_{\text{ref}}$ obtained by a uniform refinement of $\mathcal{T}_L$ from case (i) with $M_{\mathrm{Q}} = 2$ and a large index set $\mathcal{P}_{\text{ref}}$ which includes all indices generated in this experiment. The effectivity indices are plotted in Figure 7.7. Overall, they oscillate within the interval $(6, 13)$ in all cases.

7.3.3 Experiment 2 - Estimation of directional derivatives on L-shaped domain

In this experiment, we consider the parametric model problem (4.5) posed on the L-shaped domain $D = (-1,1)^2 \setminus (-1,0]^2$ and we choose the random field (5.44) used in the experiment of Section 5.4.2. In particular, we also assume here that the parameters y_m in (5.44) are the images of uniformly distributed independent mean-zero random variables on $\Gamma_m = [-1,1]$ for all $m \in \mathbb{N}$.

Similarly to previous experiment in Section 7.3.2, we choose a quantity of interest that involves the average value of a directional derivative of the primal solution over a small region away from the reentrant corner of the domain. More precisely, we set $f_0 = 1$, $\boldsymbol{f} = (0,0)$, $g_0 = 0$, and $\boldsymbol{g} = (\chi_{T_g}, 0)$, where χ_{T_g} denotes the characteristic function of the triangle

$$T_g := \operatorname{conv}\big(\{(1/2,-1),(1,-1),(1,-1/2)\}\big),$$

(see Figure 7.8 (left)), so that the functionals in (7.21) and (7.22) read as

$$F(v) = \int_\Gamma \int_D v(\mathbf{x},\mathbf{y})\,d\mathbf{x}\,d\pi(\mathbf{y}) \qquad \text{and} \qquad G(v) = -\int_\Gamma \int_{T_g} \frac{\partial v}{\partial x_1}(\mathbf{x},\mathbf{y})\,d\mathbf{x}\,d\pi(\mathbf{y}) \qquad \forall\ v \in V.$$

Note that in this example, both the primal and dual solutions exhibit a geometric singularity at the reentrant corner of the domain; see Figures 5.7(c) for a plot of the primal solution and

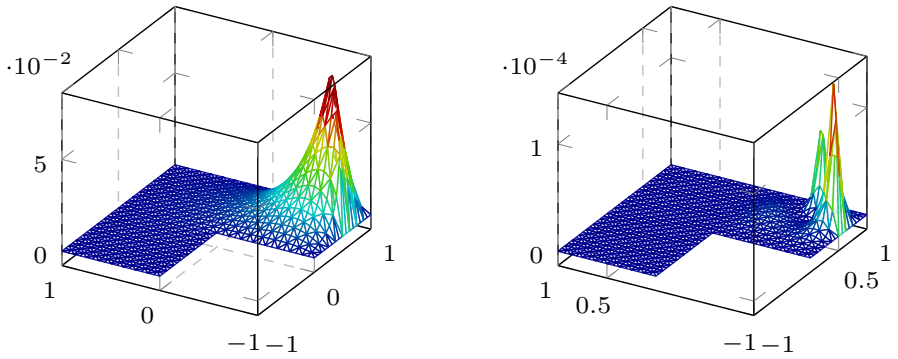


Figure 7.9. The mean field (left) and the variance (right) of the dual Galerkin solution for the model problem in Section 7.3.3.

Figure 7.9 (left) for a plot of the dual solution. In addition, notice that the dual solution exhibits also a singularity along the line connecting the points $(1/2, -1)$ and $(1, -1/2)$. Such singularity is due to a low regularity of the goal functional G.

Similarly to the experiment of Section 5.4.2, our first aim in this experiment is to show the advantages of using adaptivity in both components of Galerkin approximations for adaptive Algorithm 7.1 (cf. Figure 5.2). To this end, we consider the expansion coefficients in (5.44) with $\sigma = 2$ (corresponding to a slow decay of the amplitudes α_m), $A \approx 0.547$, and we choose $M_{\mathrm{Q}} = 1$ in (5.35). Starting with the coarse triangulation $\mathcal{T}_0$ depicted in Figure 7.8 (left) and setting the tolerance to tol = 1e-05, we run Algorithm 7.1 for six different sets of marking parameters and plot the error estimates $\tau_\ell \zeta_\ell$ computed at each iteration; see Figure 7.10.

In the cases where only one component of the Galerkin approximation is enriched (i.e., either $\theta_{\mathcal{X}} = 0$ or $\theta_{\mathcal{Y}} = 0$ as in the first two sets of parameters in Figure 7.10), the error estimates $\tau_\ell \zeta_\ell$ quickly stagnate as iterations progress, and the set tolerance cannot be reached. If both components are enriched but no adaptivity is used (i.e., $\theta_{\mathcal{X}} = \theta_{\mathcal{Y}} = 1$, see the third set of parameters in Figure 7.10), then the error estimates decay throughout all iterations. However, in this case, the overall decay rate is slow and eventually deteriorates due to the number of degrees of freedom growing very fast, in particular, during the iterations with parametric enrichments (see the filled pentagon markers in Figure 7.10). The deterioration of the decay rate is also observed for both the fourth and fifth sets of marking parameters in Figure 7.10, where adaptivity is only used for enhancing the spatial or the parametric component of approximations, respectively. The best rate is achieved for the sixth set of marking parameters, $\theta_{\mathcal{X}} = 0.2$, $\theta_{\mathcal{Y}} = 0.8$, where adaptivity is used for both components of Galerkin approximations. Thus, in agreement with results of experiment in

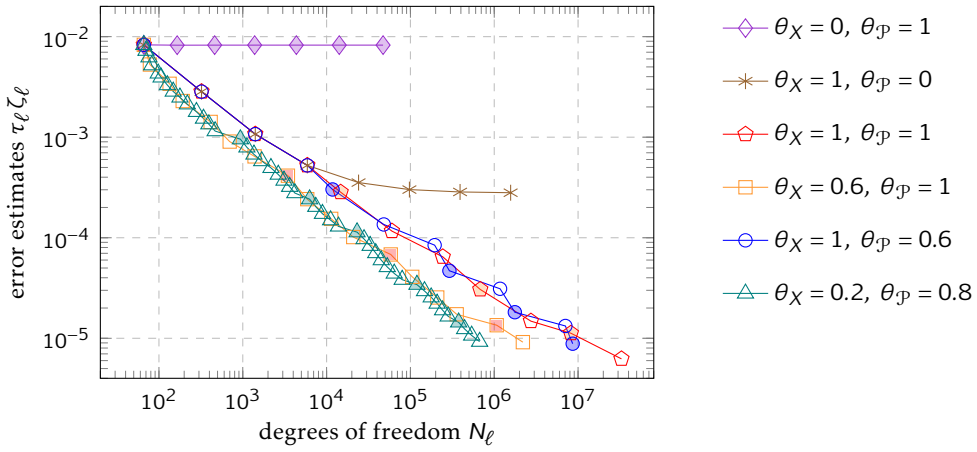


Figure 7.10. Error estimates $\tau_\ell \zeta_\ell$ at each iteration of Algorithm 7.1 for different sets of marking parameters in the numerical experiment of Section 7.3.3. Filled markers indicate iterations at which parametric enrichments occur.

Section 5.4.2, we conclude that for the same level of accuracy, adaptive enrichments in both components provide more balanced approximations with overall less degrees of freedom and lead to faster convergence rates.

Let us now run Algorithm 7.1 with the following two sets of marking parameters: (i) $\theta_X = 0.3$, $\theta_{\mathcal{P}} = 0.8$; (ii) $\theta_X = 0.15$, $\theta_{\mathcal{P}} = 0.95$. In each case, we consider the expansion coefficients in (5.44) with slow ($\sigma = 2$) and fast ($\sigma = 4$) decay of the amplitudes α_m (in the latter case, fixing $\gamma = A\zeta(\sigma) = 0.9$ results in $A \approx 0.832$). In all computations, we choose $M_{\mathcal{Q}} = 1$ in (5.35) and set the tolerance to tol = 1e-05.

Figure 7.8 (right) depicts an adaptively refined triangulation produced by Algorithm 7.1 in case (i) for the problem with slow decay of the amplitude coefficients (similar triangulations were obtained in other cases). Observe that the triangulation effectively captures spatial features of primal and dual solutions. Indeed, it is refined in the vicinity of the reentrant corner and, similarly to the experiment in Section 7.3.2, in the vicinity of points $(1/2, -1)$ and $(1, -1/2)$.

Table 7.3 collects the outputs of all computations. On the one hand, we observe that in case (i), for both slow and fast decay of the amplitude coefficients, the algorithm took fewer iterations compared to case (ii) (32 versus 57 for $\sigma = 2$ and 33 versus 57 for $\sigma = 4$) and reached the tolerance faster (see the final times t in Table 7.3). On the other hand, due to a larger θ_X in case (i), the algorithm produced more refined triangulations (see the values of $\#\mathcal{T}_L$ in Table 7.3). Also, we observe that final index sets generated for the problem with slow decay ($\sigma = 2$) are larger than those for the problem with fast decay ($\sigma = 4$) (20 indices versus 12 in case (i) and 29 indices

	$\theta_X = 0.3$, $\theta_{\mathcal{P}} = 0.8$				$\theta_X = 0.15$, $\theta_{\mathcal{P}} = 0.95$			
	$\sigma = 2$		$\sigma = 4$		$\sigma = 2$		$\sigma = 4$	
L	32		33		57		57	
t (sec)	323		367		506		485	
cost	3,017,395		2,386,455		5,502,298		3,482,930	
$\tau_L \zeta_L$	8.4228e-06		8.2214e-06		9.3225e-06		9.6034e-06	
N_L	782,100		561,384		818,989		521,645	
$\#\mathcal{T}_L$	79,029		94,446		57,203		62,102	
$\#\mathcal{P}_L$	20		12		29		17	
$M_{\mathcal{P}_L}$	6		3		6		4	
$\mathcal{P}_\ell$	$\ell = 11$	(0 1) (2 0)	$\ell = 8$	(2 0)	$\ell = 14$	(0 1) (2 0)	$\ell = 10$	(2 0)
	$\ell = 18$	(0 0 1) (1 1 0)	$\ell = 14$	(3 0)	$\ell = 25$	(0 0 1) (1 1 0) (3 0 0)	$\ell = 20$	(0 1) (3 0)
	$\ell = 22$	(0 0 0 1) (1 0 1 0) (2 1 0 0) (3 0 0 0)	$\ell = 19$	(0 1) (4 0)	$\ell = 36$	(0 0 0 1) (0 2 0 0) (1 0 1 0) (2 1 0 0) (4 0 0 0)	$\ell = 29$	(1 1) (4 0)
			$\ell = 23$	(1 1) (5 0)			$\ell = 39$	(0 0 1) (2 1 0) (5 0 0)
	$\ell = 27$	(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)	$\ell = 28$	(2 1) (6 0)	$\ell = 45$	(0 0 0 0 1) (0 1 1 0 0) (1 0 0 1 0) (1 2 0 0 0) (2 0 1 0 0) (3 1 0 0 0)	$\ell = 48$	(1 0 1) (3 1 0) (6 0 0)
	$\ell = 31$	(0 0 0 0 0 1) (0 1 1 0 0 0) (1 0 0 0 1 0) (1 2 0 0 0 0) (4 0 0 0 0 0)	$\ell = 32$	(0 0 1) (3 1 0)	$\ell = 52$	(0 0 0 0 0 1) (0 1 0 0 1 0) (0 1 0 1 0 0) (1 0 0 0 0 1) (1 0 0 0 1 0) (1 1 1 0 0 0) (2 0 0 1 0 0) (2 2 0 0 0 0) (3 0 1 0 0 0) (4 1 0 0 0 0) (5 0 0 0 0 0)	$\ell = 56$	(0 0 0 1) (2 0 1 0) (4 1 0 0) (7 0 0 0)

Table 7.3. The outputs obtained by running Algorithm 7.1 for the model problem in Section 7.3.3 with $\theta_X = 0.3$, $\theta_{\mathcal{P}} = 0.8$ (case (i)) and $\theta_X = 0.15$, $\theta_{\mathcal{P}} = 0.95$ (case (ii)) for both slow ($\sigma = 2$) and fast ($\sigma = 4$) decay of the amplitude coefficients.

versus 17 in case (ii)). Furthermore, the algorithm tends to activate more parameters and to generate polynomial approximations of lower degree for the problem with slow decay (e.g., in case (i), polynomials of total degree 4 in 6 parameters for $\sigma = 2$ versus polynomials of total degree 6 in 3 parameters for $\sigma = 4$). Note that this behaviour has been previously observed in numerical experiments for parametric problems on the square domain (see [29]).

Figure 7.11 (resp., Figure 7.12) shows the convergence history of three error estimates (τ_ℓ, ζ_ℓ, and $\tau_\ell \zeta_\ell$) and the reference error in the goal functional in case (i) (resp. case (ii)) of marking

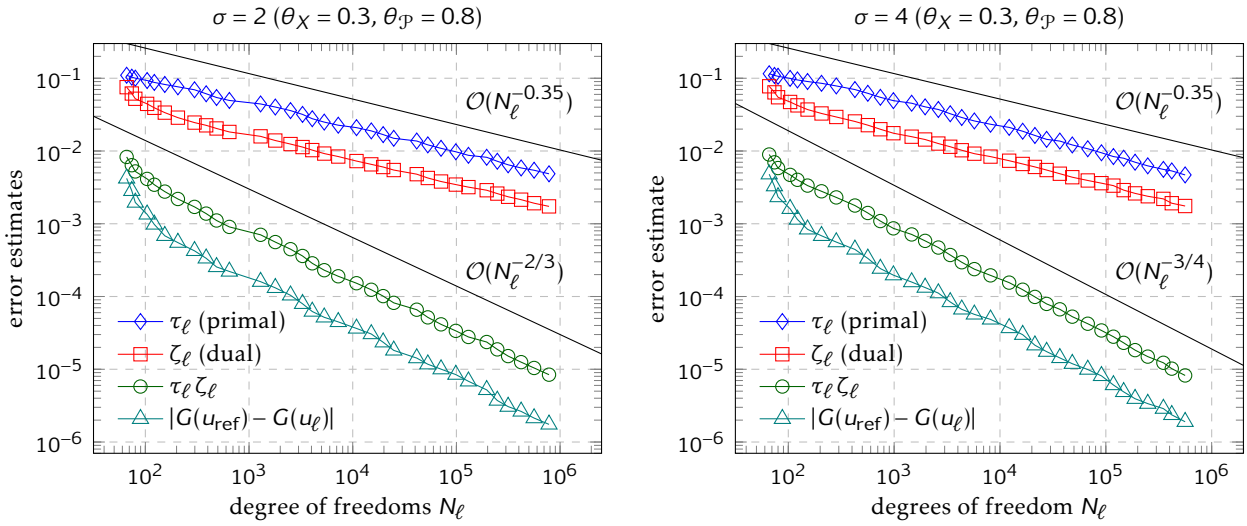


Figure 7.11. Error estimates τ_ℓ, ζ_ℓ, $\tau_\ell\zeta_\ell$ and the reference error $|G(u_{\mathrm{ref}}) - G(u_\ell)|$ at each iteration of Algorithm 7.1 with $\theta_X = 0.3$, $\theta_{\mathcal{P}} = 0.8$ (case (i)) for $\sigma = 2$ (left) and $\sigma = 4$ (right) for the model problem in Section 7.3.3. Here, $G(u_{\mathrm{ref}}) = 1.789774\text{e-}2$ for $\sigma = 2$ and $G(u_{\mathrm{ref}}) = 1.855648\text{e-}2$ for $\sigma = 4$.

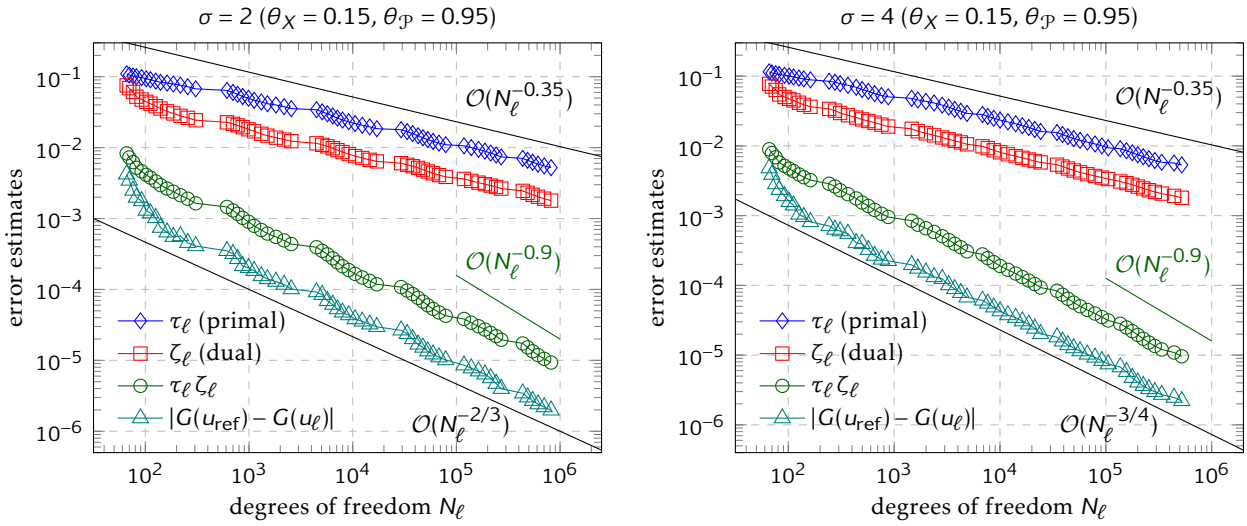


Figure 7.12. Error estimates τ_ℓ, ζ_ℓ, $\tau_\ell\zeta_\ell$ and the reference error $|G(u_{\mathrm{ref}}) - G(u_\ell)|$ at each iteration of Algorithm 7.1 with $\theta_X = 0.15$, $\theta_{\mathcal{P}} = 0.95$ (case (ii)) for $\sigma = 2$ (left) and $\sigma = 4$ (right) for the model problem in Section 7.3.3. Here, $G(u_{\mathrm{ref}}) = 1.789774\text{e-}2$ for $\sigma = 2$ and $G(u_{\mathrm{ref}}) = 1.855648\text{e-}2$ for $\sigma = 4$.

parameters. Firstly, we can see that the estimates $\tau_\ell\zeta_\ell$ converge with a faster rate for the problem with $\sigma = 4$ than for the problem with $\sigma = 2$. This is true in both cases of marking parameters. In particular, the overall convergence rate is about $\mathcal{O}(N^{-3/4})$ when $\sigma = 4$, whereas it is about $\mathcal{O}(N^{-2/3})$ when $\sigma = 2$. Secondly, we observe an improved convergence rate during mesh-refinement steps in case (ii) (i.e., for smaller θ_X and larger $\theta_{\mathcal{P}}$). For both problems with $\sigma = 2$ and $\sigma = 4$ this rate is about $\mathcal{O}(N^{-0.9})$, i.e., very close to the optimal one (see Figure 7.12).

We conclude the experiment by testing the effectivity of the goal-oriented error estimation at

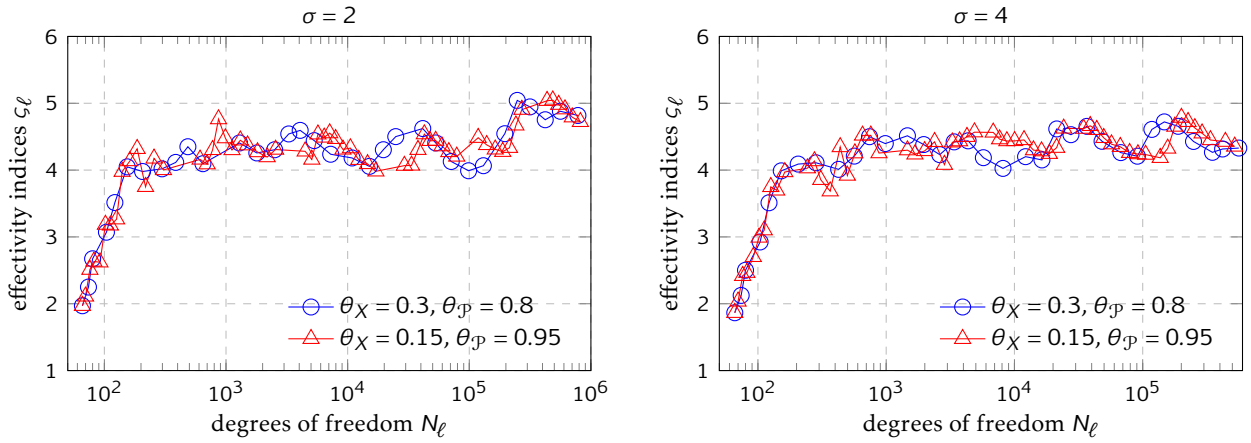


Figure 7.13. The effectivity indices for the goal-oriented error estimates $\tau_\ell\zeta_\ell$ at each iteration of Algorithm 7.1 for the model problem in Section 7.3.3.

each iteration of Algorithm 7.1. We compute the effectivity indices ς_ℓ defined in (7.23) by employing reference Galerkin solutions u_{ref} to problems with slow ($\sigma = 2$) and fast ($\sigma = 4$) decay of the amplitude coefficients. Specifically, for both problems we employ the same reference triangulation $\mathcal{T}_{\text{ref}}$ (to be the uniform refinement of $\mathcal{T}_L$ generated in case (i) for the problem with slow decay), but use two reference index sets (namely, for $\sigma = 2$, we set $\mathcal{P}_{\text{ref}} := \mathcal{P}_L$, where $\mathcal{P}_L$ is generated for the problem with slow decay in case (ii) and for $\sigma = 4$, we set $\mathcal{P}_{\text{ref}} := \mathcal{P}_L \cup \mathcal{M}_L$ with the corresponding $\mathcal{P}_L$ and $\mathcal{M}_L$ generated for the problem with fast decay in case (ii)). The computed effectivity indices are plotted in Figure 7.13. As iterations progress, they tend to concentrate within the interval $(4, 5)$ in all cases.

For the parametric model problem considered in this experiment, we conclude that Algorithm 7.1 performs better if the (spatial) marking threshold θ_X is sufficiently small and the (parametric) marking threshold $\theta_{\mathcal{P}} < 1$ is sufficiently large (see the results of experiments in case (ii)). In fact, in case (ii), the estimates $\tau_\ell\zeta_\ell$ converge with nearly optimal rates during spatial refinement steps for problems with slow and fast decay of the amplitude coefficients. Furthermore, in this case, the algorithm generates richer index sets which lead to more accurate parametric approximations.

7.3.4 Experiment 3 - Pointwise estimation on slit domain

In this last experiment, we test the performance of adaptive Algorithm 7.1 for the parametric model problem (4.5) posed on the domain $D = D_\delta$ with $D_\delta = (-1,1)^2 \setminus \overline{T}_\delta$, $\delta > 0$, where $T_\delta = \text{conv}\big(\{(0,0),(-1,\delta),(-1,-\delta)\}\big)$, introduced in Section 5.4.4. That is, we consider an approximation

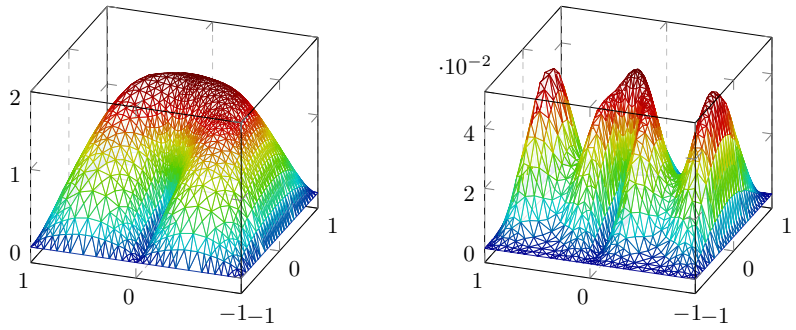


Figure 7.14. The mean field (left) and the variance (right) of the primal Galerkin solution for the model problem in Section 7.3.4.

of the slit domain $D = (-1,1)^2 \setminus ([-1,0] \times \{0\})$ as δ tends to zero. Throughout, we let $\delta = 0.005$.

Following [57], we consider a modification of parametric coefficient (5.44). For all $m \in \mathbb{N}$ and $\mathsf{x} \in D$, let $a_m(\mathsf{x})$ be the spatial functions

$$a_m(\mathsf{x}) := \alpha_m \cos(2\pi\beta_1(m)\,x_1)\cos(2\pi\beta_2(m)\,x_2), \qquad \alpha_m := Am^{-\sigma},$$

where A, σ, β_2, and β_2 are defined as in Section 5.4.2. Then, given two constants c, $\varepsilon > 0$, we define

$$a(\mathsf{x},\mathsf{y}) := \frac{c}{\alpha_{\min}} \left(\sum_{m=1}^{\infty} y_m a_m(\mathsf{x}) + \alpha_{\min} \right) + \varepsilon, \quad \mathsf{x} \in D,\ \mathsf{y} \in \Gamma, \tag{7.25}$$

where $\alpha_{\min} := A\zeta(\sigma)$ and the parameters y_m are the images of uniformly distributed independent mean-zero random variables on $\Gamma_m = [-1,1]$ for all $m \in \mathbb{N}$. It is easy to see that $a(\mathsf{x},\mathsf{y}) \in [\varepsilon, 2c+\varepsilon]$ for all $\mathsf{x} \in D$ and $\mathsf{y} \in \Gamma$. Note that (7.25) can be written in the form (4.8) by setting $a_0(x) = c + \varepsilon$ and the expansion coefficients equal to $(c\,a_m(\mathsf{x}))/\alpha_{\min}$. Furthermore, conditions (4.9) and (4.10) are satisfied with $a_0^{\min} = a_0^{\max} = c + \varepsilon$ and $\gamma = c/(c+\varepsilon)$, respectively.

It is known that solution u to problem (4.5) in this example exhibits a singularity induced by the slit in the domain (cf. the numerical experiment in Section 5.4.4). Our aim in this experiment is to approximate the (mean) value of u at some fixed point $\mathsf{x}_0 \in D$ away from the slit. To that end (and to stay within the framework of the bounded goal functional G in (7.22)), we fix a sufficiently small $r > 0$ and define g_0 as the *mollifier* (see [109]),

$$g_0(\mathsf{x}) = g_0(\mathsf{x};\mathsf{x}_0,r) := \begin{cases} C\exp\left(-\frac{r^2}{r^2-\|\mathsf{x}-\mathsf{x}_0\|_2^2}\right) & \text{if } \|\mathsf{x}-\mathsf{x}_0\|_2 < r, \\ 0 & \text{otherwise.} \end{cases} \tag{7.26}$$

Here, $\|\cdot\|_2$ denotes the Euclidean norm and C is a normalisation constant chosen such that

$$\int_D g_0(\mathsf{x})\,d\mathsf{x} = 1.$$

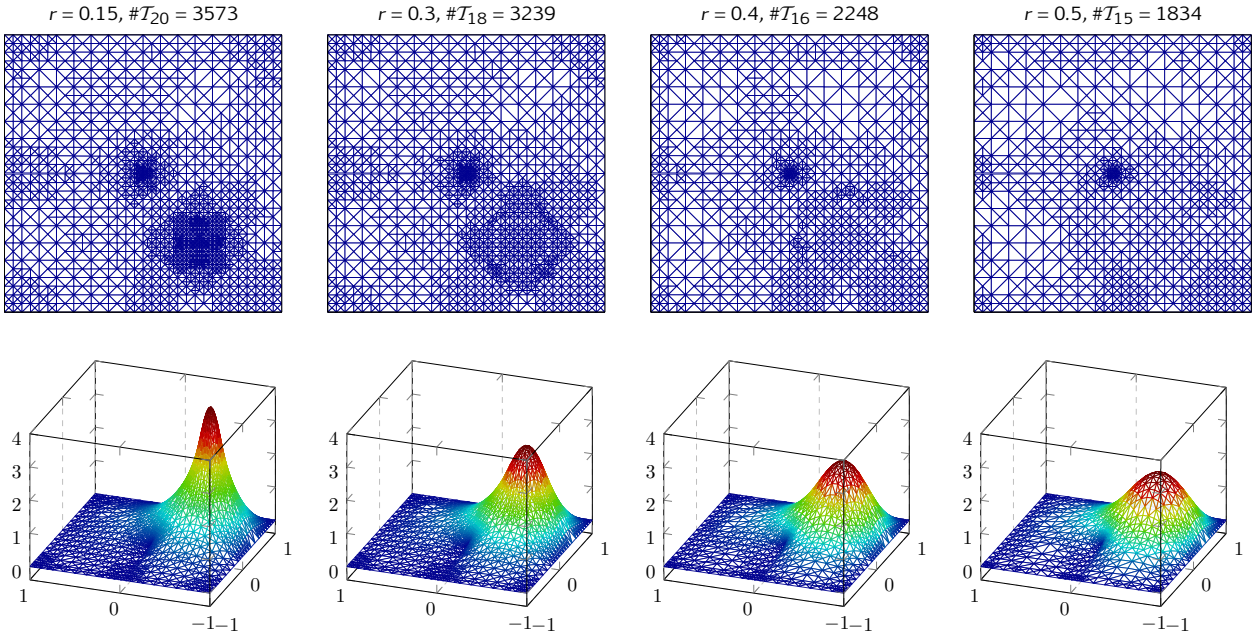


Figure 7.15. Adaptively refined triangulations (top row) and mean fields of dual Galerkin solutions (bottom row) computed using the mollifier g_0 in (7.26) with $r = 0.15, 0.3, 0.4, 0.5$, for the model problem in Section 7.3.4.

Note that the value of the constant C is independent of the location of $\mathsf{x}_0 \in D$, provided that r is chosen sufficiently small such that $\operatorname{supp}(g_0(\mathsf{x};\mathsf{x}_0,r)) \subset D$. In this case, $C \approx 2.1436\, r^{-2}$ (see, e.g., [109]).

Setting $f_0 = 1$, $\boldsymbol{f} = (0,0)$ and $\boldsymbol{g} = (0,0)$, the functionals in (7.21) and (7.22) read as

$$F(v) = \int_\Gamma \int_D v(\mathsf{x},\mathsf{y})\, d\mathsf{x}\, d\pi(\mathsf{y}) \qquad \text{and} \qquad G(v) = \int_\Gamma \int_D g_0(\mathsf{x})\, v(\mathsf{x},\mathsf{y})\, d\mathsf{x}\, d\pi(\mathsf{y}) \qquad \forall\ v \in V.$$

Note that if $u(\mathsf{x},\mathsf{y})$ is continuous in the spatial neighbourhood of x_0, then $G(u)$ converges to the mean value $\mathbb{E}[u(\mathsf{x}_0,\mathsf{y})]$ as r tends to zero.

We fix $c = 10^{-1}$, $\varepsilon = 5 \cdot 10^{-3}$, $\sigma = 2$, $A = 0.6$ and choose $\mathsf{x}_0 = (0.4, -0.5) \in D$. In all computations performed in this experiment, we use the coarse triangulation $\mathcal{T}_0$ depicted in Figure 5.11(a); Figure 7.14 shows the mean field (left) and the variance (right) of the primal Galerkin solution.

First, we fix tol = 7e-03 and run Algorithm 7.1 to compute dual Galerkin solutions for different values of radius r in (7.26). Figure 7.15 shows the refined triangulations (top row) and the corresponding mean fields of dual Galerkin solutions (bottom row) for $r = 0.15, 0.3, 0.4, 0.5$. As observed in experiments of Sections 7.3.2 and 7.3.3, the triangulations generated by the goal-oriented adaptive algorithm simultaneously captures spatial features of primal and dual solutions. In this experiment, the triangulations are refined in the vicinity of each corner, with par-

	$\theta_X = 0.3$, $\theta_{\mathcal{P}} = 0.8$		$\theta_X = 0.15$, $\theta_{\mathcal{P}} = 0.8$	
L	30		53	
t (sec)	436		646	
cost	3,047,041		4,841,526	
$\tau_L \zeta_L$	4.9335e-04		5.4956e-04	
N_L	827,421		705,915	
$\#\mathcal{T}_L$	79,518		67,871	
$\#\mathcal{P}_L$	21		21	
$M_{\mathcal{P}_L}$	6		6	
$\mathcal{P}_\ell$	$\ell = 10$	(0 1) (2 0)	$\ell = 12$	(0 1) (2 0)
	$\ell = 16$	(0 0 1) (1 1 0) (3 0 0)	$\ell = 23$	(0 0 1) (1 1 0) (3 0 0)
	$\ell = 22$	(0 0 0 1) (1 0 1 0) (2 1 0 0)	$\ell = 35$	(0 0 0 1) (1 0 1 0) (2 1 0 0)
	$\ell = 26$	(0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (4 0 0 0 0)	$\ell = 43$	(0 0 0 0 1) (1 0 0 1 0) (2 0 1 0 0) (4 0 0 0 0)
	$\ell = 29$	(0 0 0 0 0 1) (0 1 1 0 0 0) (0 2 0 0 0 0) (1 0 0 0 1 0) (2 0 0 1 0 0) (3 0 1 0 0 0) (3 1 0 0 0 0)	$\ell = 47$	(0 0 0 0 0 1) (0 1 1 0 0 0) (0 2 0 0 0 0) (1 0 0 0 1 0) (2 0 0 1 0 0) (3 0 1 0 0 0) (3 1 0 0 0 0)

Table 7.4. The outputs obtained by running Algorithm 7.1 with $\theta_X = 0.3$, $\theta_{\mathcal{P}} = 0.8$ (case (i)) and $\theta_X = 0.15$, $\theta_{\mathcal{P}} = 0.8$ (case (ii)) in the case $r = 0.15$ for the model problem in Section 7.3.4.

ticularly strong refinement near the origin, where the primal solution exhibits a singularity (see Figure 7.14 (left)); in addition to that, for smaller values of r (e.g., $r = 0.15,\ 0.3$), the triangulation is strongly refined in a neighbourhood of x_0 due to sharp gradients in the corresponding dual solutions (note that the refinements in the neighbourhood of x_0 become coarser as r increases).

Let us now fix $r = 0.15$ (which gives $C \approx 95.271$ in (7.26)) and run Algorithm 7.1 with two sets of marking parameters: (i) $\theta_X = 0.3$, $\theta_{\mathcal{P}} = 0.8$; (ii) $\theta_X = 0.15$, $\theta_{\mathcal{P}} = 0.8$. In both computations we choose $M_{\mathcal{Q}} = 1$ in (5.35) and set the tolerance tol = 6.0e-04.

In Table 7.4, we collect the outputs of computations in both cases. In agreement with results of the experiments of previous Sections 7.3.2 and 7.3.3, we see that running the algorithm with a smaller value of θ_X (i.e., in case (ii)) requires more iterations to reach the tolerance (see the values of L in both columns in Table 7.4). We also observe that, for a fixed $\theta_{\mathcal{P}}$, choosing a smaller θ_X naturally results in a less refined final triangulation ($\#\mathcal{T}_L = 67,871$ in case (ii) versus $\#\mathcal{T}_L = 79,518$

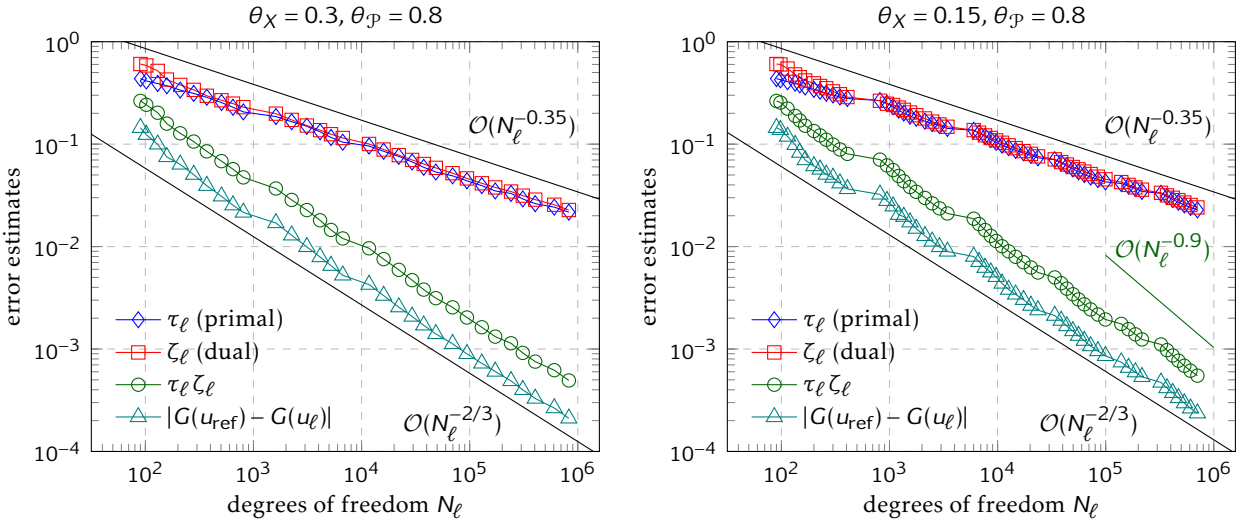


Figure 7.16. Error estimates τ_ℓ, ζ_ℓ, $\tau_\ell\zeta_\ell$ and the reference error $|G(u_{\mathrm{ref}}) - G(u_\ell)|$ at each iteration of Algorithm 7.1 with $\theta_X = 0.3$, $\theta_{\mathcal{P}} = 0.8$ (case (i), left) and $\theta_X = 0.15$, $\theta_{\mathcal{P}} = 0.8$ (case (ii), right) in the case $r = 0.15$ for the model problem in Section 7.3.4. Here, $G(u_{\mathrm{ref}}) = 0.144497\mathrm{e}{+01}$.

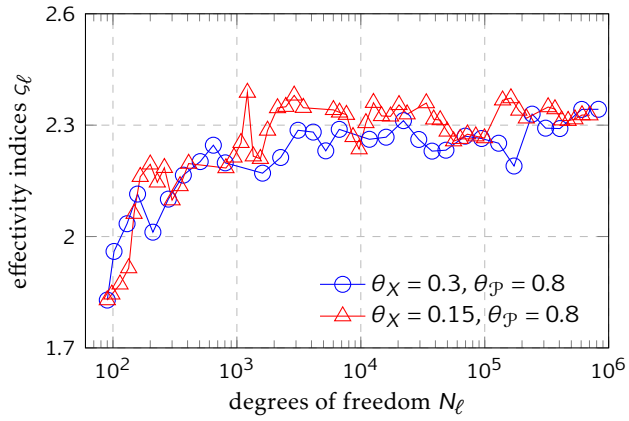


Figure 7.17. The effectivity indices for the goal-oriented error estimates $\tau_\ell\zeta_\ell$ at each iteration of Algorithm 7.1 in the case $r = 0.15$ for the model problem in Section 7.3.4.

in case (i)), although, for the chosen tolerance, the final index set $\mathcal{P}_L$ generated is the same in both cases (21 indices with 6 active parameters).

By looking now at Figure 7.16 we observe that the energy error estimates τ_ℓ and ζ_ℓ decay with the same rate of about $\mathcal{O}(N^{-0.35})$ for both sets of marking parameters; this yields an overall rate of about $\mathcal{O}(N^{-2/3})$ for $\tau_\ell\zeta_\ell$ in both cases. However, we can see that in case (ii), the estimates $\tau_\ell\zeta_\ell$ decay with a nearly optimal rate of $\mathcal{O}(N^{-0.9})$ during mesh-refinement steps. This is due to a smaller value of the marking parameter θ_X in this case and consistent with what we observed in the experiment in Section 7.3.3.

Finally, we compute the effectivity indices ς_ℓ defined in (7.23) at each iteration of the algorithm

by employing a reference Galerkin solution u_{ref} computed using the triangulation $\mathcal{T}_{\mathrm{ref}}$ (to be the uniform refinement of $\mathcal{T}_L$ produced in case (i)) and the reference index set $\mathcal{P}_{\mathrm{ref}} := \mathcal{P}_L \cup \mathcal{M}_L$, where $\mathcal{P}_L$ and $\mathcal{M}_L$ are generated in case (ii). The effectivity indices are plotted in Figure 7.17. This plot shows that the sequence $(\tau_\ell \zeta_\ell)_\ell$, $\ell = 0, \ldots, L$, provides sufficiently accurate estimates of the error in approximating $G(u)$, as the effectivity indices tend to vary in a range between 1.8 to 2.5 for both sets of marking parameters.

The results of this experiment show that Algorithm 7.1 with appropriate choice of marking parameters generates effective approximations to the mean of the quantity of interest associated with point values of the spatially singular solution to the considered parametric model problem. In agreement with results of the experiment in Section 7.3.3, we conclude that smaller values of the spatial marking parameter θ_X (such as $\theta_X = 0.15$ as in case (ii)) are, in general, preferable, as they yield nearly optimal convergence rates (for the error in the goal functional) during spatial refinement steps.

Chapter 8

Concluding remarks

The development of robust numerical schemes for efficient discretisation of continuous mathematical models with inherent uncertainties has been a very active research theme in recent years. In this context, and largely for PDE based problems, the design of efficient adaptive strategies is evermore a demanding task to mitigate the so-called 'curse of dimensionality', that is a deterioration of convergence rates and an exponential growth of the computational cost as the dimension of the discrete parameter space increases. In particular, adaptive algorithms are indispensable when solving a particularly challenging class of parametric problems represented by PDEs whose inputs data depend (e.g., in an affine way) on *infinitely* many uncertain parameters.

In this work, we considered a simple example of elliptic boundary value problem whose uncertainty is represented by a random diffusion coefficient. Furthermore, homogeneous Dirichlet conditions were imposed on the boundary of physical domains. The numerical method we primarily decided to focus on is the intrusive SGFEM whereas the type of a posteriori error estimates that we used to design adaptive strategies for our model problem are hierarchical and two-level estimates. The more important contribution of this thesis has been the design and development of innovative adaptive algorithms, under the SGFEM framework, for the numerical computation of solutions to the parametric PDE problem using the above mentioned type of error estimates.

The first adaptive algorithm presented in Chapter 5 is 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; we do not claim originality of the associated analysis, yet we observe that, to our knowledge, an efficient algorithm of this type was missing in the literature. 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 step.

The second adaptive algorithm proposed in Chapter 6, essentially follows the same lines of that of Chapter 5. For both algorithms, in fact, the aim was the estimation of the energy norm of the global error. However, in Chapter 6 we introduced the novel two-level error estimate and proved that it is both efficient and reliable. The associated algorithm using such estimate, in its four possible versions, does not require an extra step for solving any linear system for the estimation of errors arising from spatial discretisations. When compared to hierarchical error estimation, this approach leads to an undeniable benefit in terms of the overall computational time (see the numerical experiment in Section 6.4.1). Moreover, with a rigorous convergence analysis of the adaptive algorithm, the sequence of generated two-level error estimates is proved to converge to zero (see [25]), and further to that, we proved that the two versions of the algorithm which use only the Dörfler strategy in the associated marking criteria (i.e., Algorithms 6.1v1 and 6.1v2), yield a sequence of global errors which converges linearly.

Finally, we developed a goal-oriented adaptive algorithm for the approximation of a quantity of interest which is a linear functional of the solution to parametric problem. The algorithm is driven by two-level a posteriori estimates of the energy errors (although it works with any reliable and computable a posteriori error estimate) in Galerkin approximations of the primal as well as associated dual solutions. The main novelty is that the components of the error estimates for primal and dual solutions are used to guide the adaptive enhancement of the discrete space as well as to assess the error reduction in the product of these estimates which is seen to be a reliable estimate for the approximation error in the quantity of interest. This information about the error reduction is then employed to choose between spatial refinement and parametric enrichment throughout the iterations of the algorithm.

Future work that would conclude the investigation reported here, includes the mathematical justification, via convergence analysis, of the adaptive SGFEM algorithm using hierarchical estimates for the energy error estimation as well as of the goal-oriented adaptive algorithm in Chapter 7. Alternatively, other obvious possible extensions include the use of other compatible types of spatial mesh-refinement rules to be used in the proposed algorithms (e.g., general newest vertex bisections or regular refinements, instead of just longest edge bisections), the redesign of the algorithms in case of other elliptic operators, or the treatment of non-homogeneous boundary conditions. In fact, for example, we observe that the analysis of hierarchical error es-

timation for parametric PDEs is well-established in case of homogeneous Dirichlet conditions (see Sections 5.1 and 5.2) but the extension to non-homogeneous boundary conditions seems still missing; the same extension could be investigated if two-level error estimates are used in adaptive algorithms. Furthermore, the case of problems with parametric right-hand sides sources as well as parameter-dependent functions in the definition of the goal functional for goal-oriented estimation may be interesting topics that would contribute to fill gaps in the current literature about parametric PDEs.

To conclude, we recall that the numerical software implementing the proposed adaptive algorithms in this thesis is available online and can be used to reproduce the presented numerical results as well as to help future investigations about parametric elliptic problems.

Appendix A

Numerical experiment of Section 6.4.2 (extended version)

Tables A.1–A.4 collect the computational costs (6.39) as well as the empirical convergence rates for Algorithms 6.1v1–6.1v4 applied to the parametric model problem in Section 6.4.2. The empirical convergence rates are computed as the slopes of the lines which are best fit, in the least squares sense, of the overall error estimates τ_ℓ computed by the algorithm with the corresponding pair of marking parameters $(\theta_X, \theta_{\mathcal{P}}) \in \Theta \times \Theta$ with $\Theta = \{0.1, 0.2, \dots, 0.9\}$. Observe that all rates are similar and vary in a range between -0.36 and -0.32. In each table, numbers in boldface indicate the smallest cost in the corresponding row (i.e., for fixed θ_X), whereas the starred boldface number denotes the overall smallest cost in the table.

Appendix A. Numerical experiment of Section 6.4.2 (extended version)

Algorithm 6.1v1

θ_X \ $\theta_{\mathcal{P}}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	36,634,764 −0.3395	36,634,764 −0.3395	36,634,764 −0.3395	36,634,764 −0.3395	36,634,764 −0.3395	37,126,693 −0.3412	38,135,658 −0.3401	38,948,918 −0.3398	**36,522,593** −0.3401
0.2	10,652,382 −0.3386	10,652,382 −0.3386	10,652,382 −0.3386	10,652,382 −0.3386	10,652,382 −0.3386	10,472,434 −0.3401	10,611,056 −0.3392	10,842,902 −0.3395	**9,891,950** −0.3398
0.3	5,737,346 −0.3380	5,737,346 −0.3380	5,737,346 −0.3380	5,737,346 −0.3380	5,737,346 −0.3380	5,398,269 −0.3392	5,444,071 −0.3386	5,491,501 −0.3390	**4,487,527** −0.3392
0.4	4,066,841 −0.3369	4,066,841 −0.3369	4,066,841 −0.3369	4,066,841 −0.3369	4,066,841 −0.3369	3,657,156 −0.3382	3,703,037 −0.3379	3,738,567 −0.3384	**3,005,547** −0.3385
0.5	2,974,895 −0.3360	2,974,895 −0.3360	2,974,895 −0.3360	2,974,895 −0.3360	2,974,895 −0.3360	2,523,497 −0.3374	2,526,413 −0.3373	2,521,302 −0.3381	**2,193,757** −0.3380
0.6	2,838,789 −0.3371	2,838,789 −0.3371	2,838,789 −0.3371	2,838,789 −0.3371	2,838,789 −0.3371	2,323,857 −0.3383	2,351,810 −0.3384	2,331,065 −0.3389	**1,900,951** −0.3385
0.7	2,658,382 −0.3373	2,658,382 −0.3373	2,658,382 −0.3373	2,658,382 −0.3373	2,658,382 −0.3373	2,094,382 −0.3380	2,046,871 −0.3390	2,014,430 −0.3399	**1,566,530** −0.3394
0.8	2,454,929 −0.3346	2,454,929 −0.3346	2,454,929 −0.3346	2,454,929 −0.3346	2,454,929 −0.3346	2,403,912 −0.3354	1,628,563 −0.3367	**1,560,286**★ −0.3363	1,731,044 −0.3373
0.9	3,042,687 −0.3278	3,042,687 −0.3278	3,042,687 −0.3278	3,042,687 −0.3278	3,042,687 −0.3278	2,891,115 −0.3308	1,978,191 −0.3289	**1,776,192** −0.3321	1,993,972 −0.3334

Table A.1. Computational cost (top of the cell) and empirical convergence rates (bottom of the cell) for Algorithm 6.1v1 applied to the parametric model problem in Section 6.4.2.

Algorithm 6.1v2

θ_X \ $\theta_{\mathcal{P}}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	**55,591,871** −0.3543	**55,591,871** −0.3543	**55,591,871** −0.3543	61,960,516 −0.3478	71,558,237 −0.3372	82,193,108 −0.3326	90,977,432 −0.3288	99,219,735 −0.3281	> 1e+08 −0.3216
0.2	11,801,518 −0.3543	11,801,518 −0.3543	11,801,518 −0.3543	**11,606,375** −0.3542	12,864,430 −0.3479	13,991,407 −0.3424	14,616,770 −0.3377	15,930,094 −0.3376	18,204,663 −0.3362
0.3	5,385,296 −0.3499	5,385,296 −0.3499	5,385,296 −0.3499	5,385,296 −0.3499	**5,340,256** −0.3492	5,757,081 −0.3454	6,042,307 −0.3416	5,796,230 −0.3452	6,330,829 −0.3391
0.4	3,587,223 −0.3457	3,587,223 −0.3457	3,587,223 −0.3457	3,587,223 −0.3457	3,626,569 −0.3461	3,432,938 −0.3467	3,338,087 −0.3442	**3,086,323** −0.3473	3,165,582 −0.3425
0.5	2,874,852 −0.3429	2,874,852 −0.3429	2,874,852 −0.3429	2,874,852 −0.3429	2,874,852 −0.3429	2,380,185 −0.3464	2,560,036 −0.3451	**2,081,426** −0.3465	2,582,765 −0.3430
0.6	2,883,427 −0.3383	2,883,427 −0.3383	2,883,427 −0.3383	2,883,427 −0.3383	2,883,427 −0.3383	2,259,538 −0.3411	2,307,901 −0.3421	**1,764,686** −0.3422	2,078,219 −0.3415
0.7	3,157,697 −0.3292	3,157,697 −0.3292	3,157,697 −0.3292	3,157,697 −0.3292	3,157,697 −0.3292	2,146,095 −0.3350	1,973,460 −0.3383	1,966,801 −0.3389	**1,488,993**★ −0.3398
0.8	3,381,315 −0.3237	3,381,315 −0.3237	3,381,315 −0.3237	3,381,315 −0.3237	3,381,315 −0.3237	2,613,691 −0.3320	1,641,372 −0.3355	**1,549,138** −0.3369	1,720,006 −0.3378
0.9	4,886,790 −0.3153	4,886,790 −0.3153	4,886,790 −0.3153	4,886,790 −0.3153	4,886,790 −0.3153	3,708,374 −0.3205	2,288,775 −0.3246	2,071,551 −0.3282	**1,993,972** −0.3334

Table A.2. Computational cost (top of the cell) and empirical convergence rates (bottom of the cell) for Algorithm 6.1v2 applied to the parametric model problem in Section 6.4.2.

Algorithm 6.1v3

θ_X \ $\theta_{\mathcal{P}}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	37,126,693 −0.3412	38,436,652 −0.3392	**31,766,942** −0.3391	38,948,918 −0.3398	40,891,821 −0.3400	35,855,809 −0.3397	30,252,882 −0.3397	44,306,077 −0.3389	47,582,801 −0.3342
0.2	10,472,434 −0.3401	10,293,846 −0.3388	8,743,434 −0.3388	10,842,902 −0.3395	10,790,957 −0.3397	9,833,369 −0.3395	**8,317,634** −0.3395	12,082,564 −0.3389	12,942,155 −0.3338
0.3	5,398,269 −0.3392	5,386,660 −0.3382	4,609,593 −0.3384	5,491,501 −0.3390	5,194,711 −0.3390	4,573,863 −0.3391	**4,270,672** −0.3390	6,113,283 −0.3382	5,957,047 −0.3334
0.4	3,657,156 −0.3382	3,573,880 −0.3372	3,169,527 −0.3373	3,738,567 −0.3384	3,352,712 −0.3384	3,024,178 −0.3380	**2,634,872** −0.3378	4,146,897 −0.3370	3,567,033 −0.3325
0.5	2,523,497 −0.3374	2,380,561 −0.3367	2,459,900 −0.3368	2,521,302 −0.3381	2,102,539 −0.3379	2,260,210 −0.3374	**1,847,454** −0.3371	2,721,150 −0.3369	2,572,804 −0.3320
0.6	2,323,857 −0.3384	2,168,310 −0.3377	2,271,816 −0.3382	2,331,065 −0.3389	1,828,574 −0.3384	2,004,779 −0.3390	**1,533,861** −0.3386	2,528,526 −0.3378	2,294,306 −0.3325
0.7	2,094,382 −0.3380	1,891,752 −0.3375	1,970,087 −0.3376	2,014,430 −0.3399	**1,496,851**★ −0.3393	1,710,029 −0.3383	1,793,937 −0.3383	2,185,402 −0.3377	1,837,025 −0.3325
0.8	2,403,912 −0.3354	2,162,469 −0.3347	2,240,383 −0.3362	**1,560,286** −0.3363	1,645,652 −0.3370	1,940,368 −0.3368	2,048,616 −0.3370	1,620,466 −0.3353	2,089,746 −0.3297
0.9	2,891,115 −0.3308	2,621,348 −0.3295	2,830,679 −0.3283	**1,776,192** −0.3321	1,885,067 −0.3329	2,470,591 −0.3282	2,619,845 −0.3285	1,880,106 −0.3295	2,611,297 −0.3231

Table A.3. Computational cost (top of the cell) and empirical convergence rates (bottom of the cell) for Algorithm 6.1v3 applied to the parametric model problem in Section 6.4.2.

Algorithm 6.1v4

θ_X \ $\theta_{\mathcal{P}}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	**65,375,862** −0.3482	78,893,946 −0.3367	82,752,064 −0.3375	86,536,330 −0.3383	85,276,880 −0.3389	94,093,402 −0.3378	> 1e+08 −0.3339	> 1e+08 −0.3285	> 1e+08 −0.3235
0.2	**11,852,403** −0.3516	12,956,995 −0.3447	13,658,830 −0.3427	14,138,022 −0.3475	14,888,877 −0.3441	16,118,678 −0.3443	17,014,103 −0.3412	17,774,020 −0.3370	22,821,433 −0.3294
0.3	5,393,465 −0.3482	**5,187,233** −0.3503	5,976,264 −0.3465	5,607,496 −0.3462	6,018,687 −0.3443	6,200,892 −0.3415	6,737,413 −0.3383	6,628,919 −0.3354	8,667,208 −0.3279
0.4	3,359,537 −0.3466	2,993,784 −0.3499	**2,968,892** −0.3484	3,086,323 −0.3473	3,280,115 −0.3460	3,526,098 −0.3446	3,229,531 −0.3419	3,942,973 −0.3367	5,087,723 −0.3298
0.5	2,380,185 −0.3464	2,317,914 −0.3467	2,570,641 −0.3466	**2,081,426** −0.3465	2,294,857 −0.3456	2,461,136 −0.3442	2,727,436 −0.3422	2,513,794 −0.3384	3,221,702 −0.3316
0.6	2,259,538 −0.3411	2,163,842 −0.3402	1,719,454 −0.3413	1,764,686 −0.3422	1,897,407 −0.3423	2,067,075 −0.3412	**1,672,508** −0.3395	1,935,515 −0.3370	2,563,621 −0.3309
0.7	2,146,095 −0.3350	1,952,007 −0.3347	2,000,424 −0.3363	1,966,801 −0.3389	**1,460,210**★ −0.3383	1,604,638 −0.3389	1,740,662 −0.3386	2,050,900 −0.3370	1,855,200 −0.3305
0.8	2,613,691 −0.3320	2,613,691 −0.3304	2,429,679 −0.3331	**1,549,138** −0.3368	1,634,584 −0.3375	1,806,369 −0.3381	1,977,211 −0.3380	1,621,561 −0.3349	2,093,208 −0.3304
0.9	3,708,374 −0.3205	3,151,928 −0.3189	3,183,738 −0.3249	2,071,551 −0.3288	1,885,067 −0.3329	2,470,591 −0.3287	2,386,770 −0.3311	**1,880,106** −0.3295	2,439,044 −0.3254

Table A.4. Computational cost (top of the cell) and empirical convergence rates (bottom of the cell) for Algorithm 6.1v4 applied to the parametric model problem in Section 6.4.2.

Appendix B

Stochastic T-IFISS package

In this appendix, we describe the *Stochastic T-IFISS* package [28] used for all numerical experiments in this thesis. The name T-IFISS stands for *Triangular Incompressible Flow & Iterative Solver Software*. Stochastic T-IFISS is the extension, to stochastic Galerkin approximations of diffusion problems with random coefficients, of the core version of the open source Matlab toolbox T-IFISS [123] for solving deterministic elliptic problems on two-dimensional domains using finite element method; spatial discretisations in both Stochastic T-IFISS and T-IFISS are performed by means of triangular meshes. In turn, T-IFISS has been developed on top of the Matlab toolbox IFISS [60] that was produced to perform computational experiments in the monograph [61]; in IFISS, deterministic elliptic problems on domains are discretised using rectangular meshes. In what follows, we briefly highlight the main components of Stochastic T-IFISS. Note that this is not supposed to be a detailed technical description of the toolbox. This short appendix is rather intended to be the starting point of a future complete documentation of the software.

B.1 Overview. Stochastic T-IFISS has been mainly designed to support the investigation about the topics reported in this work. The package is organised in a modular way using task-specific modules, each of them dedicated to the various component aspects of solving individual parametric PDE problems. This makes easy for the user to experiment with the code and, most importantly, to include additional features as well as extract parts of the package to be used for distinct purposes. In particular, the toolbox is accessible to anyone with a basic knowledge of Matlab. Also, Stochastic T-IFISS takes advantage of Matlab high-level programming, portability, and readability as well as vectorisation features that enable fast and efficient computation: wherever possible, Stochastic T-IFISS routines are written so as to exploit vectorised computation over, for example, finite elements (such as assembling of stiffness matrices or mesh-refinements).

The Stochastic T-IFISS toolbox includes all directories composing the T-IFISS package. In fact, Stochastic T-IFISS makes use of several T-IFISS files such as marking strategies and quadrature rules. Most important features of current version 1.2 of Stochastic T-IFISS are Galerkin linear and quadratic spatial discretisations (quadratic discretisations only for non-adaptive codes), visualisation of generated domains as well as mean value and variance of computed solutions, fast mesh-refinement routines (spatial refinement can be either element or midpoint based), and implementation of adaptive algorithms driven by built-in a posteriori error estimates of the generated solutions.

Main routines in Stochastic T-IFISS implementing stochastic Galerkin approximations for parametric PDEs problems have been initially developed for the numerical SGFEM discretisation of model problem (4.5) (see [27]); a further extension included all routines for the goal-oriented numerical approximation of quantities of interest discussed in [26]. In particular, Stochastic T-IFISS contains self-adaptive test problems implementing adaptive Algorithms 5.1 and 7.1 able to reproduce the experiments in Sections 5.4 and 7.3. Both hierarchical (see Chapter 5) and two-level (see Chapter 6) error estimates are then also implemented and can be used for a posteriori energy error estimations.

B.2 Directory structure and test problems. In Stochastic T-IFISS, function files to run SGFEM approximations for parametric PDEs are included in the directory `stoch_diffusion`. The names of all function files in `stoch_diffusion` start with `stoch_`. Within this directory, main subdirectories are:

- `/stoch_diffusion_adapt/` includes important routines used by self-adaptive examples, such as main drivers for the setup of initial parameters, marking strategies, and error estimation techniques;
- `/stoch_goafem/` includes all routines for self-adaptive goal-oriented examples; the names of all function files in this subdirectory start with `stoch_goafem_`;
- `/test_problem/` contains files for the definition of diffusion coefficients, boundary conditions, and source terms for the set up of built-in reference test problems. These include problems with both estimation of the energy error and prescribed quantities of interest.

Current reference problems implemented in Stochastic T-IFISS are parametric PDEs posed over three available spatial domains, square, L-shaped, and slit domains (see Section 5.4), and using

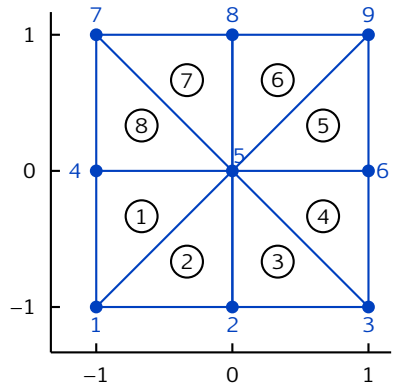
three main type of parametric coefficients. These are the synthetic random field (5.44) from [54, 55], the KL-expansion of random fields with covariance (3.9), and expansion (3.17) from [88]. For goal-oriented examples, the setup of reference test problems is the one described in Section 7.3.1; this includes the parametric version of Example 7.3 in [95] for the estimation of directional derivatives as well as the estimation in approximating pointwise values (see Section 7.3). Main drivers runnig self-adaptive algorithms for the above mentioned model problems are `stoch_adapt_testproblem` and `stoch_goafem_testproblem`, respectively, that can be found inside the `/test_problem/` subdirectory.

B.3 Data structures (spatial approximations). Let $\mathcal{T} = \{T_1, \dots, T_{N_\mathcal{T}}\}$ be a conforming and shape-regular triangulation with $N_\mathcal{T} := \#\mathcal{T}$ elements of $D \subset \mathbb{R}^2$. Let $N_\mathcal{X} := \#\mathcal{N}(\mathcal{T})$ be the number of total vertices, $N_\mathcal{X}^D := \#\mathcal{N}^\circ(\mathcal{T})$ be the number of interior vertices, and $N_\mathcal{X}^{\partial D} := \#(N_\mathcal{X} \setminus N_\mathcal{X}^D)$ be the number of boundary vertices (note that $N_\mathcal{X} = N_\mathcal{X}^D + N_\mathcal{X}^{\partial D}$). Let $\mathcal{N}(\mathcal{T}) = \{\mathsf{x}_1, \dots, \mathsf{x}_{N_\mathcal{X}}\}$ be the set of total vertices. The triangulation $\mathcal{T}$ is represented by the matrices `xy` and `evt`, and vectors `interior` and `bound`:

- `xy` is a $N_\mathcal{X} \times 2$ matrix containing the physical coordinates of all vertices. The i-th row of `xy` stores the coordinates of the i-th vertex $\mathsf{x}_i = (x_1^{(i)}, x_2^{(i)}) \in \mathcal{N}(\mathcal{T})$, i.e., `xy(i,:)` $= [x_1^{(i)}\ x_2^{(i)}]$;
- `evt` is a $N_\mathcal{T} \times 3$ matrix containing the elements vertices' numbers. That is, the n-th element $T_n = \mathrm{conv}(\{\mathsf{x}_i, \mathsf{x}_j, \mathsf{x}_k\}) \in \mathcal{T}$ with vertices $\mathsf{x}_i, \mathsf{x}_j, \mathsf{x}_k \in \mathcal{N}(\mathcal{T})$ is stored in `evt(n,:)` $= [$`i j k`$]$, with vertices counted in counterclockwise order;
- `interior` is a $N_\mathcal{X}^D \times 1$ vector containing the global numbers of interior vertices of $\mathcal{T}$. For example, if $\mathcal{N}^\circ(\mathcal{T}) = \{\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_4, \mathsf{x}_6\}$, then `interior` $= [1\ 2\ 4\ 6]^T$;
- `bound` is a $N_\mathcal{X}^{\partial D} \times 1$ vector containing the global numbers of boundary vertices of $\mathcal{T}$. For example, if $\mathcal{N}(\mathcal{T}) \cap \partial D = \{\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_5\}$, then `bound` $= [1\ 2\ 5]^T$. Note that $\{$`interior, bound`$\} = \{1, 2, \dots, N_\mathcal{X}\}$.

Figure B.1 shows an illustrative triangulation and the associated matrices and vectors described above; these are, in particular, the spatial data structures required for the numerical computation of the SGFEM solution. A remark about enumeration of vertices and edges is due here. Given $T = \mathrm{conv}(\{\mathsf{x}_i, \mathsf{x}_j, \mathsf{x}_k\}) \in \mathcal{T}$, with 1-st vertex x_i, 2-nd vertex x_j, and 3-rd vertex x_k counted counterclockwise, it is assumed that

- the 1-st edge of T is the one in front of x_i, i.e., $\mathrm{conv}(\{\mathsf{x}_j, \mathsf{x}_k\})$;



xy

1	−1	−1
2	0	−1
3	1	−1
4	−1	0
5	0	0
6	1	0
7	−1	1
8	0	1
9	1	1

evt

1	5	4	1
2	1	2	5
3	5	2	3
4	3	6	5
5	5	6	9
6	9	8	5
7	5	8	7
8	7	4	5

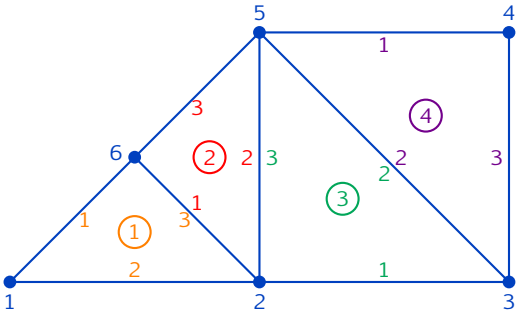
bound

1	1
2	2
3	3
4	4
5	6
6	7
7	8
8	9

interior

1	5

Figure B.1. Triangulation $\mathcal{T}$ of the square domain $D = (-1,1)^2$ with $N_{\mathcal{T}} = 8$ elements (circled numbers in black) specified by `evt` matrix and $N_X = 9$ vertices (numbers in blue) specified by `xy` matrix. Interior and boundary vertices are stored in `interior` and `bound` vectors, respectively.



eboundt

1	1	1
2	1	2
3	2	3
4	3	1
5	4	1
6	4	3

els

1	1.00	1.41	1.00
2	1.00	1.41	1.00
3	1.41	2.00	1.41
4	1.41	2.00	1.41

eex

1	0	0	2
2	1	3	0
3	0	4	2
4	0	3	0

tve

1	0	0	1
2	3	3	0
3	0	2	2
4	0	2	0

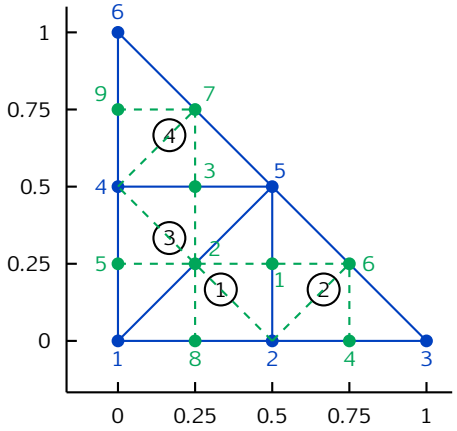
Figure B.2. Local enumeration of the edges of a conforming and shape-regular (structured) triangulation $\mathcal{T}$ with four elements; numbers in blues are vertices' global numbers of $\mathcal{T}$ whereas circled numbers are elements' numbers. Coloured numbers denote local elements' edges. Associated data structures `eboundt`, `els`, `eex` and `tve`. Zeros entries in `eex` and `tve` indicate the boundary edges.

- the 2-nd edge of T is the one in front of x_j, i.e., $\mathrm{conv}(\{\mathsf{x}_k, \mathsf{x}_i\})$;
- the 3-rd edge of T is the one in front of x_k, i.e., $\mathrm{conv}(\{\mathsf{x}_i, \mathsf{x}_j\})$.

In particular, for a shape-regular (*structured*) triangulation $\mathcal{T}$, the local enumeration of the vertices of an element $T \in \mathcal{T}$ is such that the longest edge of T is the second one (see Figure B.2).

Additional data structures are the `eboundt`, `els`, `eex`, and `tve` matrices:

- `eboundt` is a matrix containing the boundary elements' numbers and local numbers of the corresponding edges lying on the boundary ∂D. By *boundary element*, we mean an element having either one or two edges lying on the boundary;
- `els` is a $N_{\mathcal{T}} \times 3$ matrix containing, on each row, the lengths of the 3 edges of all elements in $\mathcal{T}$;
- `eex` is a $N_{\mathcal{T}} \times 3$ matrix storing the element patches (see (2.7)) of all elements in $\mathcal{T}$. That is, if



xyY

1	0.50	0.25
2	0.25	0.25
3	0.25	0.75
4	0.75	0.00
5	0.00	0.25
6	0.75	0.25
7	0.25	0.75
8	0.25	0.00
9	0.00	0.75

evtY

1	1	2	8
2	4	6	1
3	5	2	3
4	3	7	9

boundY

1	4
2	5
3	6
4	7
5	8
6	9

Ybasis

1	1	2	1	3
2	1	3	2	2
3	3	4	3	1
4	2	2	1	1
5	3	3	1	1
6	2	2	2	2
7	4	4	2	2
8	1	1	3	3
9	4	4	3	3

Figure B.3. Data structures for the detail grid associated with the detail space Y on a given triangulation $\mathcal{T}$ with $\#\mathcal{T} = 4$ elements.

$\omega(T_n) = \{T_m, T_\ell, T_k\}$ with T_m sharing the 1-st edge of T_n, T_ℓ sharing the 2-nd edge of T_n, and T_k sharing the 3-rd edge of T_n, then `eex(n,:) = [m ℓ k]`;

- `tve` is a $N_{\mathcal{T}} \times 3$ matrix for the 'edge-location' of neighbours in the element patches. For example, if $T_n \in \mathcal{T}$ shares the 1-st, the 3-rd, and the 2-nd edge of its corresponding neighbours (stored in `eex`), respectively, then `tve(n,:) = [1 3 2]`.

See Figure B.2 for an example of the data structures `eboundt`, `els`, `eex`, and `tve` for a given triangulation. In particular, these four data structures are needed for solving local element residual problems (5.34) for the computation of hierarchical error estimate $\eta_{X\mathcal{P}}$ (see `stoch_diffpost_p1_yp` below).

Given the finite element space $X(\mathcal{T})$ associated with $\mathcal{T}$, Stochastic T-IFISS also includes some data structures to store information about the *detail grid* for the corresponding first-order detail space Y (see (5.3)) associated with uniform NVB refinements of $\mathcal{T}$ (see Figure 5.1(b)). That is, the detail grid stores information about the 'mesh' consisting of midpoints introduced by uniform NVB refinements associated with Y. Let $N_Y = \#\mathcal{E}(\mathcal{T})$ and $N_Y^{\partial D} = \#(\mathcal{E}(\mathcal{T}) \setminus \mathcal{E}^\circ(\mathcal{T}))$ be the number of total and boundary midpoints, respectively. The detail grid is represented by the matrices `xyY`, `evtY`, `boundY`, and `Ybasis`, where:

- `xyY` is a $N_Y \times 2$ matrix containing the physical coordinates of the midpoints of all edges of $\mathcal{T}$ (cf. `xy`);
- `evtY` is a $N_{\mathcal{T}} \times 3$ matrix containing all midpoints' numbers per element. That is, if midpoints $z_i, z_j, z_k \in \mathcal{N}^+$ lie on the 1-st, the 2-nd, and the 3-rd edge of the n-th element $T_n \in \mathcal{T}$,

respectively, then `evtY(n,:) = [i j k]`;

- `boundY` is a $N_Y^{\partial D} \times 1$ matrix containing the global numbers of boundary midpoints (cf. `bound`);
- `Ybasis` is a $N_Y \times 4$ matrix storing information about position of linear basis functions of Y, that is, information about midpoint positions with respect to the elements. For example, suppose that the i-th row of `Ybasis` is `Ybasis(i,:) = [n m j k]`. It means that midpoint z_i lies on the (i-th) edge shared by elements $T_n, T_m \in \mathcal{T}$, and that such edge is the j-th (local) edge of T_n and the k-th (local) edge of T_m (i.e., $j, k \in \{1,2,3\}$). In particular, the first two entries `n` and `m`, indicate the patch of (the two) elements containing the support of the i-th basis function of Y (i.e., $\mathrm{supp}(\psi_i) \subseteq T_n \cup T_m$).

See Figure B.3 for an illustrative example triangulation with associated detail grid data structures `xyY`, `evtY`, `boundY`, and `Ybasis`. These four matrices are used by Stochastic T-IFISS routines for the assembling of the two-level error estimate $\tau_{X\mathcal{P}}$ (see `stoch_diffpost_p1_yp_2level` below).

B.4 Data structures (parametric approximations). Let $\mathcal{P}$ be the finite index set for parametric approximations, and recall that $N_{\mathcal{P}}$ and $M_{\mathcal{P}}$ denote its cardinality and the number of active parameters, respectively. In Stochastic T-IFISS, two main variables take account of the number of parameters: the variable `norv`, is a global variable storing the maximum number of parameters allowed for numerical computation, while `noarv` denotes the number of active parameters (i.e., $M_{\mathcal{P}}$) which vary throughout adaptive computations. Other main variables and data structures are:

- `distribution` is the variable storing the distribution type of random parameters. These can be uniformly distributed on $[-1,1]$ or having 'truncated' Gaussian density (4.27) on $[-1,1]$;
- `indset` is a $N_{\mathcal{P}} \times M_{\mathcal{P}}$ matrix which stores the given index set $\mathcal{P}$. Each row of `indset` is an index $\nu \in \mathcal{P}$;
- `G` is a $1 \times (M_{\mathcal{P}}+1)$ cell array which stores the G matrices (4.47) for the given index set `indset`;
- `Q_indset` is a matrix storing the finite detail index set $\mathcal{Q}$, defined in (5.35), associated with `indset`;
- `GPQ` is the cell array which stores the G matrices associated with index sets $\mathcal{P}$ and $\mathcal{Q}$ (cf. (4.47)); these matrices appear in the assembling of the right-hand side of (5.27) where there are both $u_{X\mathcal{P}} \in V_{X\mathcal{P}}$ and $v \in X \otimes \mathcal{P}_\mu$, with $\mu \in \mathcal{Q}$.

B.5 List of main functions. Here, we report an incomplete list of functions of Stochastic T-IFISS. In particular, we list the main functions for the stages composing the loop of self-adaptive routines. We briefly explain what each function does (notice that in writing functions' names, we do not specify the input and output arguments). For more information, we refer to detailed descriptions supplied by single functions in the package.

Initialisation step:

- `stoch_adapt_init_param`: scriptfile that sets up all initial variables. Among the most important, there are variables for the version's type of the adaptive algorithm, tolerance, type of error estimate (hierarchical or two-level), marking strategy (maximum or Dörfler), and associated threshold parameters;
- `stoch_adapt_init_spatial`: scriptfile for the generation of the initial triangulation of the spatial domain and all associated data structures (see Section B.3);
- `stoch_adapt_init_stoch`: scriptfile for the set up of random diffusion coefficients, distribution of parameters as well as associated data structures for parametric discretisations (see Section B.4).

The same scriptfiles for the goal-oriented adaptive algorithm (e.g., `stoch_goafem_init_param`) fulfill similar tasks.

SOLVE module:

- `stoch_femp1_setup`: function for assembling (first-order) stochastic matrices as well as source vectors in linear system (4.44);
- `stoch_femp2_setup`: as `stoch_femp1_setup` but for second-order spatial Galerkin approximations;
- `stoch_impose_bcx`: imposes prescribed Dirichlet boundary conditions on both sides of linear system (4.44);
- `stoch_est_minresx`: function implementing a preconditioned MINRES solver for the solution of linear system (4.44) and the computation of SGFEM approximations (see [122]).

ESTIMATE module:

- `stoch_diffpost_p1_yp`: this function computes the energy norm (5.37) of spatial estimator $e_{Y\mathcal{P}}$ defined in (5.20) by implementing the element residual method (5.34);

- `stoch_diffpost_p1_yp_linsys`: this function also computes the energy norm of spatial estimator $e_{\mathcal{YP}}$ defined in (5.20) but by solving the global assembled linear system directly arising from discrete formulation (5.20) (assembled linear system is solved using Matlab backslash (\) operator);
- `stoch_diffpost_p1_yp_2level`: this function assembles the spatial part (6.31) contributing to the two-level estimate (6.33);
- `stoch_adapt_diffpost_p1_xq`: this function computes the energy norm of parametric estimator $e_{\mathcal{XQ}}$ defined in (5.21) by exploiting the decomposition into contributing individual indices in the detail index set $\mathcal{Q}$ (see (5.27)); this is used by both hierarchical and two-level error estimates.

MARK module:

- `stoch_adapt_marking`: main driver for the marking step of adaptive loops. It calls the marking strategies routines for spatial and stochastic component of SGFEM approximation; for spatial marking, the function allows the marking of both elements and midpoints (i.e., edges);
- `marking_strategy_fa`: function implementing the maximum and the Dörfler marking strategies (see Section 2.3.1);
- `get_all_marked_elem`: this function returns the set of overall marked elements (resp. midpoints) that have to be refined (resp. introduced) to keep the conformity once the current triangulation is refined.

REFINE module:

- `mesh_ref`: main driver for spatial mesh-refinements. It returns the updated data structures `xy`, `evt`, `bound`, `interior`, and `eboundt` (see Section B.3) for the refined triangulation;
- `bisection`: function implementing NVB refinements where reference edges are the longest edges;
- `stoch_pol_enrich`: scriptfile performing the parametric enrichment, i.e., it enlarges the current index set `indset` by appending the set of marked indices.

B.6 Some useful commands. Here, we report some useful Matlab commands that can be used to obtain information on the underlying triangulation as soon as spatial data structures are available (see Section B.3):

- get the number of elements and total vertices of triangulation $\mathcal{T}$, respectively:

```
size(evt,1),    size(xy,1);
```

- get the number of total edges (i.e., midpoints) of triangulation $\mathcal{T}$:

```
size(evtY,1);
```

- get the coordinates of interior and boundary vertices, respectively:

```
xy(interior,:),    xy(bound,:);
```

- get the elements sharing the vertex $\mathtt{x}_i \in \mathcal{N}(\mathcal{T})$ (i.e., the vertex patch $\omega(\mathtt{x}_i)$):

```
find( sum( (evt==i), 2) );
```

- get the elements sharing the midpoint $\mathtt{z}_j \in \mathcal{N}^+$ (i.e., the edge $E_j \in \mathcal{E}(\mathcal{T})$):

```
Ybasis(j,1:2)    or    find( sum( (evtY==j), 2) );
```

- get the elements sharing one edge with the element $T_n \in \mathcal{T}$ (i.e., the element patch $\omega(T_n)$):

```
eex(n,:)    or    setdiff( Ybasis( evtY(n,:),1:2 ), n ).
```

List of references

[1] R. A. Adams and J. J.F. Fournier, *Sobolev spaces*, Academic press, 2 ed., 2003.

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

[3] R. C. Almeida and J. T. Oden, *Solution verification, goal-oriented adaptive methods for stochastic advection–diffusion problems*, Comput. Methods Appl. Mech. and Engrg., 199 (2010), pp. 2472–2486.

[4] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, *The FEniCS project version 1.5*, Arch. Num. Soft., 3 (2015), pp. 9–23.

[5] I. Babuška, F. Nobile, and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM J. Numer. Anal., 45 (2007), pp. 1005–1034.

[6] I. Babuška and W. C. Rheinboldt, *A-posteriori error estimates for the finite element method*, Int. J. Num. Meth. Engrg., 12 (1978), pp. 1597–1615.

[7] I. Babuška and W. C. Rheinboldt, *Error estimates for adaptive finite element computations*, SIAM J. Numer. Anal., 15 (1978), pp. 736–754.

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

[9] I. Babuška, R. Tempone, and E. Zouraris, *Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 1251–1294.

[10] I. Babuška and M. Vogelius, *Feedback and adaptive finite element solution of one-dimensional boundary value problem*, Numer. Math., 44 (1984), pp. 75–102.

[11] W. Bangerth and R. Rannacher, *Adaptive finite element methods for differential equations*, Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel, 2003.

[12] R. E. Bank, *Hierarchical bases and the finite element method*, Acta Numer., 5 (1996), pp. 1–43.

[13] R. E. Bank, A. Parsania, and S. Sauter, *Saturation estimates for hp-finite element methods*, Comput. Visual. Sci., 16 (2013), pp. 195–217.

[14] R. E. Bank and A. H. Sherman, *An adaptive, multi-level method for elliptic boundary value problems*, Computing, 26 (1981), pp. 91–105.

[15] R. E. Bank and K. Smith, *A posteriori error estimates based on hierarchical bases*, SIAM J. Numer. Anal., 30 (1993), pp. 921–935.

[16] R. E. Bank and A. Weiser, *Some a posteriori error estimators for elliptic partial differential equations*, Math. Comp., 44 (1985), pp. 283–301.

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

[18] A. Barth, C. Schwab, and N. Zollinger, *Multi-Level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients*, Numer. Math., 119 (2011), pp. 123–161.

[19] H. Bauer, *Probability Theory*, de Gruyter Studies in Mathematics 23, Walter de Gruyter & Co., Berlin, New York, 1996.

[20] R. Becker, E. Estecahandy, and D. Trujillo, *Weighted marking for goal-oriented adaptive finite element methods*, SIAM J. Numer. Anal., 49 (2011), pp. 2451–2469.

[21] R. Becker and R. Rannacher, *A feed-back approach to error control in finite element methods: Basic analysis and examples*, East West J. Numer. Math., 4 (1996), pp. 237–264.

[22] R. Becker and R. Rannacher, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numer., 10 (2001), pp. 1–102.

[23] C. Berg, *Moment problems and polynomial approximation*, in Annales de la Faculté des sciences de Toulouse: Mathématiques, vol. 5, Université Paul Sabatier, Institut de Mathématiques, 1996, pp. 9–32.

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

[25] A. Bespalov, D. Praetorius, L. Rocchi, and M. Ruggeri, *Convergence of adaptive stochastic Galerkin FEM*. Preprint available at https://arxiv.org/abs/1811.09462, 2018.

[26] A. Bespalov, D. Praetorius, L. Rocchi, and M. Ruggeri, *Goal-oriented error estimation and adaptivity for elliptic PDEs with parametric or uncertain inputs*, Comput. Methods Appl. Mech. Engrg., 345 (2019), pp. 951–982.

[27] A. Bespalov and L. Rocchi, *Efficient adaptive algorithms for elliptic PDEs with random data*, SIAM/ASA J. Uncertain. Quantif., 6 (2018), pp. 243–272.

[28] A. Bespalov and L. Rocchi, *Stochastic T-IFISS*, February 2019. Available online at http://web.mat.bham.ac.uk/A.Bespalov/software/index.html#stoch_tifiss.

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

[30] M. Bieri, R. Andreev, and C. Schwab, *Sparse tensor discretization of elliptic SPDEs*, SIAM J. Sci. Comput., 31 (2009), pp. 4281–4304.

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

[32] F. A. Bornemann, B. Erdmann, and R. Kornhuber, *A posteriori error estimates for elliptic problems in two and three space dimensions*, SIAM J. Numer. Anal., 33 (1996), pp. 1188–1204.

[33] J.-M. Bourinet, *FERUM 4.1 user's guide*, Institute Français de Mécanique Avancée (IFMA), Clermont-Ferrand, France, (2010).

[34] D. Braess, *Finite Elements: Theory, fast solvers, and applications in solid mechanics*, Cambridge University Press, 2007.

[35] S. Brenner and R. Scott, *The Mathematical Theory of Finite Element Methods*, vol. 15, Springer-Verlag New York, 2008.

[36] C. Bryant, S. Prudhomme, and T. Wildey, *Error decomposition and adaptivity for response surface approximations from PDEs with parametric uncertainty*, SIAM/ASA J. Uncertain. Quantif., 3 (2015), pp. 1020–1045.

[37] T. Butler, C Dawson, and C. Wildey, *A posteriori error analysis of stochastic differential equations using polynomial chaos expansions*, SIAM J. Sci. Comput., 33 (2011), pp. 1267–1291.

[38] R. E. Caflisch, *Monte Carlo and quasi-Monte Carlo methods*, Acta Numer., 7 (1998), pp. 1–49.

[39] C. Carstensen, D. Gallistl, and J. Gedicke, *Justification of the saturation assumption*, Numer. Math., 134 (2016), pp. 1–25.

[40] J. M. Cascon, R. H. Kreuzer, R. H. Nochetto, and K. G. Siebert, *Quasi-optimal convergence rate for an adaptive finite element method*, SIAM J. Numer. Anal., 46 (2008), pp. 2524–2550.

[41] E. W. Cheney and W. A. Light, *A course in approximation theory*, vol. 101 of Graduate Studies in Mathematics, American Mathematical Soc., Providence, Rhode Island, 2009.

[42] P. G. Ciarlet, *The finite element method for elliptic problems*, North-Holland, Amsterdam, New York, Oxford, 1978.

[43] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Comput. Visual. Sci., 14 (2011), pp. 3–15.

[44] A. Cohen, R. DeVore, and C. Schwab, *Convergence rates of best N-term galerkin approximations for a class of elliptic sPDEs*, Found. Comput. Math., 10 (2010), pp. 615–646.

[45] A. Cohen, R. DeVore, and C. Schwab, *Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's*, Anal. Appl., 9 (2011), pp. 11–47.

[46] A. J. Crowder and C. E. Powell, *CBS constants & their role in error estimation for stochastic Galerkin Finite Element Methods*, J. Sci. Comput., 77 (2018), pp. 1030–1054.

[47] A. J. Crowder, C. E. Powell, and A. Bespalov, *Efficient adaptive multilevel stochastic Galerkin approximation using implicit a posteriori error estimation*. Preprint available at https://arxiv.org/abs/1806.05987v1, 2018.

[48] M. K. Deb, *Solution of Stochastic Partial Differential Equations (SPDEs) using Galerkin method: Theory and applications*, PhD thesis, University of Texas, Austin, 2000.

[49] M. K. Deb, I. Babuška, and J. T. Oden, *Solution of stochastic partial differential equations using Galerkin finite element techniques*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 6359–6372.

[50] P. Deuflhard, P. Leinen, and H. Yserentant, *Concepts of an adaptive hierarchical finite element code*, IMPACT Comput. Sci. Engin., 1 (1989), pp. 3–35.

[51] J. Dick, F. Y. Kuo, and I. H. Sloan, *High-dimensional integration: the quasi-Monte Carlo way*, Acta Numer., 22 (2013), pp. 133–288.

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

[53] W. Dörfler and R. H. Nochetto, *Small data oscillation implies the saturation assumption*, Numer. Math., 91 (2002), pp. 1–12.

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

[55] M. Eigel, C. J. Gittelson, C. Schwab, and E. Zander, *A convergent adaptive stochastic Galerkin finite element method with quasi-optimal spatial meshes*, ESAIM Math. Model. Numer. Anal., 49 (2015), pp. 1367–1398.

[56] 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.

[57] M. Eigel, C. Merdon, and J. Neumann, *An adaptive multilevel Monte Carlo method with stochastic bounds for quantities of interest with uncertain data*, SIAM/ASA J. Uncertain. Quantif., 4 (2016), pp. 1219–1245.

[58] M. Eigel and E. Zander, *ALEA – A python framework for spectral methods and low-rank approximations in uncertainty quantification*. Available online at https://bitbucket.org/aleadev/alea/src.

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

[60] H. C. Elman, A. Ramage, and D. J. Silvester, *IFISS: A computational laboratory for investigating incompressible flow problems*, SIAM Rev., 56 (2014), pp. 261–273.

[61] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, 2006.

[62] C. Erath, S. Funken, P. Goldenits, and D. Praetorius, *Simple error estimators for the Galerkin BEM for some hypersingular integral equation in 2D*, Appl. Anal., 92 (2013), pp. 1194–1216.

[63] O. G. Ernst, A. Mugler, H.-J. Starkloff, and E. Ullmann, *On the convergence of generalized polynomial chaos expansions*, ESAIM Math. Model. Numer. Anal., 46 (2012), pp. 317–339.

[64] O. G. Ernst, C. E. Powell, D. J. Silvester, and E. Ullmann, *Efficient solvers for a linear Galerkin mixed formulation of diffusion problems with random data*, SIAM J. Sci. Comput., 31 (2009), pp. 1427–1447.

[65] O. G. Ernst and E. Ullmann, *Stochastic Galerkin matrices*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1848–1872.

[66] M. Feischl, T. Führer, G. Mitscha-Eibl, D. Praetorius, and E. P. Stephan, *Convergence of adaptive BEM and adaptive FEM-BEM coupling for estimators without h-weighting factor*, Comput. Meth. App. Math., 14 (2014), pp. 485–508.

[67] M. Feischl, D. Praetorius, and K. G. van der Zee, *An abstract analysis of optimal goal-oriented adaptivity*, SIAM J. Numer. Anal., 54 (2016), pp. 1423–1448.

[68] P. Frauenfelder, C. Schwab, and R. A. Todor, *Finite elements for elliptic problems with stochastic coefficients*, Comput. Methods Appl. Mech. and Engrg., 194 (2005), pp. 205–228.

[69] S. Funken, D. Praetorius, and P. Wissgott, *Efficient implementation of adaptive P1-FEM in Matlab*, Comput. Methods Appl. Math., 11 (2011), pp. 460–490.

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

[71] R. G. Ghanem and R. M. Kruger, *Numerical solution of spectral stochastic finite element systems*, Comput. Methods Appl. Mech. Engrg., 129 (1996), pp. 289–303.

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

[73] M. B. Giles, *Multilevel Monte Carlo methods*, Acta Numer., 24 (2015), pp. 259–328.

[74] M. B. Giles and E. Süli, *Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality*, Acta Numer., 11 (2002), pp. 145–236.

[75] C. J. Gittelson, *Stochastic Galerkin approximation of operator equations with infinite dimensional noise*, in Research report/Seminar für Angewandte Mathematik, no. 10, ETH Zurich, 2011.

[76] C. J. Gittelson, *An adaptive stochastic Galerkin method for random elliptic operators*, Math. Comput., 82 (2013), pp. 1515–1541.

[77] C. J. Gittelson, *Convergence rates of multilevel and sparse tensor approximations for a random elliptic PDE*, SIAM J. Numer. Anal., 51 (2013), pp. 2426–2447.

[78] P. Grisvard, *Singularities in boundary value problems*, vol. 22 of Research in Applied Mathematics, Masson, Paris; Springer-Verlag, Berlin, 1992.

[79] M. D. Gunzburger, C. G. Webster, and G. Zhang, *Stochastic finite element methods for partial differential equations with random input data*, Acta Numer., 23 (2014), pp. 521–650.

[80] A.-L. Haji-Ali, F. Nobile, and R. Tempone, *Multi-index Monte Carlo: when sparsity meets sampling*, Numer. Math., 132 (2016), pp. 767–806.

[81] M. Holst and S. Pollock, *Convergence of goal-oriented adaptive finite element methods for nonsymmetric problems*, Numer. Methods Part. D. E., 32 (2016), pp. 479–509.

[82] A. Khan, A. Bespalov, C. E. Powell, and D. J. Silvester, *Robust a posteriori error estimation for stochastic Galerkin formulations of parameter-dependent linear elasticity equations*. Preprint available at https://arxiv.org/abs/1810.07440, 2018.

[83] H. F. King and M. Dupuis, *Numerical integration using Rys polynomials*, J. Comput. Phys., 21 (1976), pp. 144–165.

[84] I. Kossaczky, *A recursive approach to local mesh refinement in two and three dimensions*, J. Comput. Appl. Math., 55 (1995), pp. 275–288.

[85] W. A. Light and E. W. Cheney, *Approximation theory in tensor product spaces*, vol. 1169 of Lecture notes in mathematics, Springer-Verlag Berlin Heidelberg, 1985.

[86] M. Loève, *Probability Theory I*, Grad. Text in Math. 45, Springer-Verlag, New York, 4th ed., 1977.

[87] M. Loève, *Probability Theory II*, Grad. Text in Math. 46, Springer-Verlag, New York, 4th ed., 1978.

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

[89] S. Marelli and B. Sudret, *UQLab: A framework for uncertainty quantification in Matlab*, in Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management, 2014, pp. 2554–2563.

[90] L. Mathelin and O. Le Maître, *Dual-based a posteriori error estimate for stochastic finite element methods*, Comm. App. Math. Com. Sc., 2 (2007), pp. 83–115.

[91] H. G. Matthies and A. Keese, *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 1295–1331.

[92] R. E. Megginson, *An introduction to Banach space theory*, Grad. Text in Math. 183, Springer-Verlag, New York, 1998.

[93] K. Mekchay and R. H. Nochetto, *Convergence of adaptive finite element methods for general second order linear elliptic PDEs*, SIAM J. Numer. Anal., 43 (2005), pp. 1803–1827.

[94] W. F. Mitchell, *A comparison of adaptive refinement techniques for elliptic problems*, ACM Trans. Math. Software, 15 (1989), pp. 326–347.

[95] M. S. Mommer and R. Stevenson, *A goal-oriented adaptive finite element method with convergence rates*, SIAM J. Numer. Anal., 47 (2009), pp. 861–886.

[96] P. Morin, R. H. Nochetto, and K. G. Siebert, *Data oscillation and convergence of adaptive FEM*, SIAM J. Numer. Anal., 38 (2000), pp. 466–488.

[97] P. Morin, R. H. Nochetto, and K. G. Siebert, *Convergence of adaptive finite element methods*, SIAM rev., 44 (2002), pp. 631–658.

[98] P. Morin, K. G. Siebert, and A. Veeser, *A basic convergence result for conforming adaptive finite elements*, Math. Models Methods Appl. Sci., 18 (2008), pp. 707–737.

[99] P. Mund and E. P. Stephan, *An adaptive two-level method for the coupling of nonlinear FEM-BEM equations*, SIAM J. Numer. Anal., 36 (1999), pp. 1001–1021.

[100] P. Mund, E. P. Stephan, and J. Weisse, *Two-level methods for the single layer potential in* $\mathbb{R}^3$, Computing, 60 (1998), pp. 243–266.

[101] F. Nobile, L. Tamellini, F. Tesei, and R. Tempone, *An adaptive sparse grid algorithm for elliptic PDEs with lognormal diffusion coefficient*, in Sparse Grids and Applications–Stuttgart 2014, Springer, 2016, pp. 191–220.

[102] F. Nobile, R. Tempone, and C. G. Webster, *An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2411–2442.

[103] F. Nobile, R. Tempone, and C. G. Webster, *A sparse grid collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2309–2345.

[104] R. H. Nochetto, K. G. Siebert, and A. Veeser, *Theory of adaptive finite element methods: an introduction*, in Multiscale, nonlinear and adaptive approximation, Springer, 2009, pp. 409–542.

[105] R. H. Nochetto and A. Veeser, *Primer of Adaptive Finite Element Methods*, in Multiscale and Adaptivity: Modeling, Numerics and Applications, vol. 2040, Springer-Verlag Berlin Heidelberg, 2012, pp. 125–225.

[106] M. F. Pellissetti and R. G. Ghanem, *Iterative solution of systems of linear equations arising in the context of stochastic finite elements*, Adv. Eng. Softw., 31 (2000), pp. 607–616.

[107] C. E. Powell and H. C. Elman, *Block-diagonal preconditioning for spectral stochastic finite-element systems*, IMA J. Numer. Anal., 29 (2009), pp. 350–375.

[108] C. E. Powell, D. Silvester, and V. Simoncini, *An efficient reduced basis solver for stochastic Galerkin matrix equations*, SIAM J. Sci. Comput., 39 (2017), pp. A141–A163.

[109] S. Prudhomme and J. T. Oden, *On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors*, Comput. Methods Appl. Mech. and Engrg., 176 (1999), pp. 313–331.

[110] I. Pultarová, *The strengthened C.B.S. inequality constant for second order elliptic partial differential operator and for hierarchical bilinear finite element functions*, Appl. Math., 50 (2005), pp. 323–329.

[111] M. Reed and B. Simon, *Methods of modern mathematical physics*, Scientific Computation, Academic Press Inc., New York, 2nd ed., 1980.

[112] F. Riesz and B. Sz-. Nagy, *Functional Analysis*, Dover Publications Inc., New York, 1990.

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

[114] R. Y. Rubinstein and D. P. Kroese, *Simulation and Monte-Carlo method*, Pure and Applied Mathematics (New York), Wiley, 3rd ed., 2017.

[115] R. A. Ryan, *Introduction to tensor products of Banach spaces*, Springer Monographs in Mathematics, Springer-Verlag London Ltd., London, 2002.

[116] R. P. Sagar and V. H. Smith Jr, *On the calculation of Rys polynomials and quadratures*, Int. J. Quantum. Chem., 42 (1992), pp. 827–836.

[117] A. Schmidt and K. G. Siebert, *Design of adaptive finite element software. The finite element toolbox ALBERTA*, vol. 42 of Lec. Notes Comput. Sci. Eng., Springer-Verlag, Berlin, 2005.

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

[119] C. Schwab and R. A. Todor, *Karhunen–Loève approximation of random fields by generalized fast multipole methods*, J. Comput. Phys., 217 (2006), pp. 100–122.

[120] E. G. Sewell, *Automatic generation of triangulations for piecewise polynomial approximation*, PhD thesis, Purdue Univ., 1972.

[121] K. G. Siebert, *Mathematically founded design of adaptive finite element software*, in Multiscale and Adaptivity: Modeling, Numerics and Applications, vol. 2040, Springer-Verlag Berlin Heidelberg, 2012, pp. 227–309.

[122] D. Silvester and Pranjal, *An optimal solver for linear systems arising from stochastic FEM approximation of diffusion equations with random coefficients*, SIAM/ASA J. Uncertain. Quantif., 4 (2016), pp. 298–311.

[123] D. J. Silvester, A. Bespalov, Q. Liao, and L. Rocchi, *Triangular IFISS (T-IFISS) version 1.2*. Available online at https://personalpages.manchester.ac.uk/staff/david.silvester/-ifiss/tifiss.html, December 2018.

[124] R. Stevenson, *The completion of locally refined simplicial partitions created by bisection*, Math. Comp., 77 (2008), pp. 227–241.

[125] G. Strang and G. Fix, *An analysis of the finite element method*, Wellesley-Cambridge Press, Wellesley, MA, second ed., 2008.

[126] A. L. Teckentrup, R. Scheichl, M. B. Giles, and E. Ullmann, *Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients*, Numer. Math., 125 (2013), pp. 569–600.

[127] E. Ullmann, *A Kronecker product preconditioner for stochastic Galerkin finite element discretizations*, SIAM J. Sci. Comput., 32 (2010), pp. 923–946.

[128] E. Ullmann, H. C. Elman, and O. G. Ernst, *Efficient iterative solvers for stochastic Galerkin discretizations of log-transformed random diffusion problems*, SIAM J. Sci. Comput., 34 (2012), pp. A659–A628.

[129] E. Ullmann and C. E. Powell, *Solving log-transformed random diffusion problems by stochastic Galerkin mixed finite element methods*, SIAM/ASA J. Uncertain. Quantif., 3 (2015), pp. 509–534.

[130] R. Verfürth, *A posteriori error estimation and adaptive mesh-refinement techniques*, J. Comput. Appl. Math., 50 (1994), pp. 67–83.

[131] R. Verfürth, *A posteriori error estimation techniques for Finite Elements Methods*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2013.

[132] X. Wan and G. E. Karniadakis, *Error control in multi-element generalized polynomial chaos method for elliptic problems with random coefficients*, Commun. Comput. Phys., 5 (2009), pp. 793–820.

[133] N. Wiener, *The homogeneous chaos*, Amer. J. Math., 60 (1938), pp. 897–936.

[134] D. Xiu and G. E. Karniadakis, *Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 4927–4948.

[135] D. Xiu and G. E. Karniadakis, *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2002), pp. 619–644.

[136] E. Zander, *SGLib v0.9*. Available online at https://github.com/ezander/sglib.

[137] O. C. Zienkiewicz, J. P. De S. R. Gago, and D. W. Kelly, *The hierarchical concept in finite element analysis*, Comput. Struct., 16 (1983), pp. 53–65.

[138] O. C. Zienkiewicz, D. W. Kelly, J. Gago, and I. Babuška, *Hierarchical finite element approaches, error estimates and adaptive refinement*, tech. report, Maryland Univ. College Park Inst. for Physical Science and technology, 1981.