**Weierstraß-Institut**
**für Angewandte Analysis und Stochastik**
**Leibniz-Institut im Forschungsverbund Berlin e. V.**

Preprint ISSN 2198-5855

# An adaptive stochastic Galerkin tensor train discretization for randomly perturbed domains

Martin Eigel[1], Manuel Marschall[1], Michael Multerer[2]

submitted: December 21, 2018

[1] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: martin.eigel@wias-berlin.de
manuel.marschall@wias-berlin.de

[2] Institute of Computational Science
Via Giuseppe Buffi 136
6900 Lugano
Switzerland
E-Mail: michael.multerer@usi.ch

No. 2566
Berlin 2018

2010 *Mathematics Subject Classification.* 35R60, 47B80, 60H35, 65C20, 65N12, 65N22, 65J10.

*Key words and phrases.* Partial differential equations with random coefficients, tensor representation, tensor train, uncertainty quantification, stochastic finite element methods, log-normal, adaptive methods, ALS, low-rank, reduced basis methods.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

# An adaptive stochastic Galerkin tensor train discretization for randomly perturbed domains

Martin Eigel, Manuel Marschall, Michael Multerer

ABSTRACT. A linear PDE problem for randomly perturbed domains is considered in an adaptive Galerkin framework. The perturbation of the domain's boundary is described by a vector valued random field depending on a countable number of random variables in an affine way. The corresponding Karhunen-Loève expansion is approximated by the pivoted Cholesky decomposition based on a prescribed covariance function. The examined high-dimensional Galerkin system follows from the domain mapping approach, transferring the randomness from the domain to the diffusion coefficient and the forcing. In order to make this computationally feasible, the representation makes use of the modern tensor train format for the implicit compression of the problem. Moreover, an a posteriori error estimator is presented, which allows for the problem-dependent iterative refinement of all discretization parameters and the assessment of the achieved error reduction. The proposed approach is demonstrated in numerical benchmark problems.

## 1. INTRODUCTION

Uncertainties in the data for mathematical models are found naturally when dealing with real-world applications in science and engineering. Being able to quantify such uncertainties can greatly improve the relevance and reliability of computer simulations and moreover provide valuable insights into statistical properties of quantities of interest (QoI). This is one of the main motivations for the thriving field of Uncertainty Quantification (UQ).

In the application considered in this work, the computational domain is assumed as randomly perturbed. This e.g. can be an appropriate model to incorporate production tolerances into simulations and extract statistical information about how such uncertainties get transported through the assumed model. Random domain problems have been examined before, see for instance [2, 16, 25]. Often, sampling approaches are used to evaluate QoI as e.g. has been investigated with a multilevel quadrature for the the domain mapping method in [16]. As an alternative, we propose to employ a stochastic Galerkin FEM (SGFEM) to obtain a functional representation of the stochastic solution on the reference domain, which can then be used to evaluate statistical quantities. For the discretization, a Legendre polynomial chaos basis and first order FE are chosen. The expansion of the perturbation vector field in a (finite) countable sequence of random variables gives rise to a high-dimensional coupled algebraic system, which easily becomes intractable to numerical methods or results in very slow convergence. A way to overcome this problem is to utilize model order reduction techniques. In this work, we make use of the modern tensor train (TT) format [21], which provides an efficient hierarchical tensor representation and is able to exploit low-rank properties of the problem at hand. Another important technique to reduce computational complexity is the use of an adaptive discretization. In our case, this is based on a reliable a posteriori error estimator, afforded by the quasi-orthogonal approximation obtained by the SGFEM. With the described error estimator, an iterative adaptive selection of optimal discretization parameters (steering mesh refinement, anisotropic polynomial chaos and tensor ranks) is possible.

For the Karhunen-Loéve expansion of the random vector field, we employ the pivoted Cholesky decomposition derived in [14, 15]. The random coefficient and right-hand side due to the integral transformation are tackled with a tensor reconstruction method. All evaluations are carried out in the TT format, which in particular allows for the efficient computation of the error estimator as part of the adaptive algorithm.

DOI 10.20347/WIAS.PREPRINT.2566

The paper is structured as follows: The next section introduces the setting and the required assumptions of the random linear model problem. In particular, a description of the perturbation vector field and the variable transformation is given, converting the random domain problem to a random coefficient problem. Section 4 defines the Galerkin finite element discretization of the random coefficient problem in Legendre chaos polynomials. Moreover, the framework for residual based a posteriori error estimation is described. The tensor train format used for the efficient computation of the problem is introduced in Section 5. Section 6 lays out the refinement strategy for the Galerkin method, which is based on the evaluation of a reliable a posteriori error estimate in the tensor representation and an appropriate adaptive algorithm. Numerical examples are discussed in Section 7.

## 2. Diffusion problems on random domains

In this section, we formulate the stationary diffusion problem on random domains as introduced in [16]. Let $(\Omega, \mathcal{A}, \mathbb{P})$ denote a complete and separable probability space with $\sigma$-algebra $\mathcal{A}$ and probability measure $\mathbb{P}$. Here, complete means that $\mathcal{A}$ contains all $\mathbb{P}$-null sets. Moreover, for a given Banach space $\mathcal{X}$, we introduce the Lebesgue-Bochner space $L^p_{\mathbb{P}}(\Omega; \mathcal{X})$, $1 \leq p \leq \infty$, which consists of all equivalence classes of strongly measurable functions $v\colon \Omega \to \mathcal{X}$ with bounded norm

$$\|v\|_{L^p_{\mathbb{P}}(\Omega;\mathcal{X})} := \begin{cases} \left(\displaystyle\int_\Omega \|v(\cdot,\omega)\|^p_{\mathcal{X}} \,\mathrm{d}\mathbb{P}(\omega)\right)^{1/p}, & p < \infty \\ \displaystyle\operatorname*{ess\,sup}_{\omega\in\Omega} \|v(\cdot,\omega)\|_{\mathcal{X}}, & p = \infty. \end{cases}$$

We remark that if $p = 2$ and $\mathcal{X}$ is a separable Hilbert space then the Lebesgue-Bochner space $L^p_{\mathbb{P}}(\Omega; \mathcal{X})$ is isomorphic to the tensor product space $\mathcal{X} \otimes \mathcal{Y}$ with $\mathcal{Y} = L^2_{\mathbb{P}}(\Omega)$. For a comprehensive exposition of Lebesgue-Bochner spaces, we refer to [17].

In this article, we are interested in computing quantities of interest of the solution to the elliptic diffusion problem

$$\begin{aligned} -\operatorname{div}\big(\nabla u(\omega)\big) &= f \quad \text{in } D(\omega), \\ u(\omega) &= 0 \quad \text{on } \partial D(\omega), \end{aligned} \tag{1}$$

for $\mathbb{P}$-almost every $\omega \in \Omega$. We remark that it is also possible to consider non-trivial diffusion coefficients or boundary data, see e.g. [10] for the treatment of non-homogenous Dirichlet data and [20] for random diffusion coefficients. However, we emphasize that, in order to derive regularity results that allow for the data sparse approximation of quantities of interest, the data have to be analytic functions, cf. [16].

In order to guarantee the well posedness of (50), we assume that all data, i.e. the diffusion coefficient $a$ and the loading $f$ are defined with respect to the *hold-all domain*

$$\mathcal{D} := \bigcup_{\omega\in\Omega} D(\omega).$$

In order to model the random domain, we employ the concept of random vector fields. To that end, we assume that there exists a reference domain $D_{\mathrm{ref}} \subset \mathbb{R}^d$ for $d = 2, 3$ with Lipschitz continuous boundary $\partial D_{\mathrm{ref}}$ and of a random vector field

$$\boldsymbol{V}\colon D_{\mathrm{ref}} \times \Omega \to \mathbb{R}^d$$

such that $D(\omega) = \boldsymbol{V}(D_{\mathrm{ref}}, \omega)$. In addition, we require that $\boldsymbol{V}$ is a *uniform $C^1$-diffeomorphism*, i.e. there exists a constant $C_{\mathrm{uni}} > 1$ such that

$$\|\boldsymbol{V}(\omega)\|_{C^1(\overline{D_{\mathrm{ref}}};\mathbb{R}^d)}, \|\boldsymbol{V}^{-1}(\omega)\|_{C^1(\overline{D_{\mathrm{ref}}};\mathbb{R}^d)} \leq C_{\mathrm{uni}} \quad \text{for } \mathbb{P}\text{-a.e. } \omega \in \Omega. \tag{2}$$

In particular, since $\mathbf{V} \in L^\infty\big(\Omega; C^1(\overline{D_{\mathrm{ref}}})\big) \subset L^2\big(\Omega; C^1(\overline{D_{\mathrm{ref}}})\big)$, the random vector field $\boldsymbol{V}$ exhibits a Karhunen-Loève expansion of the form

$$\boldsymbol{V}(\hat{x},\omega) = \mathbb{E}[\boldsymbol{V}](\hat{x}) + \sum_{k=1}^{\infty} \boldsymbol{V}_k(\hat{x}) Y_k(\omega). \tag{3}$$

Herein, the expectation is given in terms of the Bochner integral

$$\mathbb{E}[\boldsymbol{V}](\hat{x}) := \int_\Omega \boldsymbol{V}(\hat{x},\omega)\,\mathrm{d}\mathbb{P}(\omega).$$

Note that, here and in the following, we denote material coordinates by uppercase characters, i.e. $\hat{x} \in D_{\mathrm{ref}}$, in contrast to spatial coordinates $x \in D(\omega)$. In particular, there holds $x = \boldsymbol{V}(\hat{x},\omega)$ for some $\hat{x} \in D_{\mathrm{ref}}$. The anisotropy which is induced by the spatial contributions $\{\boldsymbol{V}_k\}_k$, describing the fluctuations around the nominal value $\mathbb{E}[\boldsymbol{V}](\hat{x})$, is encoded by

$$\gamma_k := \|\boldsymbol{V}_k\|_{W^{1,\infty}(D_{\mathrm{ref}};\mathbb{R}^d)}. \tag{4}$$

In our model, we shall also make the following common assumptions.

**Assumption 2.1.**

*(i) The random variables $\{Y_k\}_k$ take values in $\Gamma_1 := [-1,1]$.*
*(ii) The random variables $\{Y_k\}_k$ are independent and identically distributed.*
*(iii) The sequence $\{\gamma_k\}_k$ is at least in $\ell^1(\mathbb{N})$.*

In view of this assumption, the Karhunen-Loève expansion (3) can always be computed if the expectation $\mathbb{E}[\boldsymbol{V}]$ and the matrix-valued covariance function

$$\mathrm{Cov}[\boldsymbol{V}](\hat{x},\hat{x}') := \int_\Omega \big(\boldsymbol{V}(\hat{x},\omega) - \mathbb{E}[\boldsymbol{V}](\hat{x})\big)\big(\boldsymbol{V}(\hat{x}',\omega) - \mathbb{E}[\boldsymbol{V}](\hat{x}')\big)\,\mathrm{d}\mathbb{P}(\omega)$$

are known. To that end, the spectral decomposition of the integral operator associated to the covariance function has to be computed. The spectral decomposition can efficiently be computed by means of the pivoted Cholesky decomposition, if the covariance function is sufficiently smooth, cf. [14, 15].

By an appropriate reparametrization, we can always guarantee that

$$\mathbb{E}[\boldsymbol{V}](\hat{x}) = \hat{x}.$$

Moreover, if we identify the random variables by their image $\boldsymbol{y} \in \Gamma := [-1,1]^{\mathbb{N}}$, we optain the representation

$$\boldsymbol{V}(\hat{x},\boldsymbol{y}) = \hat{x} + \sum_{k=1}^{\infty} \boldsymbol{V}_k(\hat{x}) y_k. \tag{5}$$

The Jacobian of $\boldsymbol{V}$ with respect to the spatial variable $x$ is accordingly given by

$$\boldsymbol{J}(\hat{x},\boldsymbol{y}) = \boldsymbol{I} + \sum_{k=1}^{\infty} \boldsymbol{V}_k'(\hat{x}) y_k.$$

Introducing the parametric domains $D(\boldsymbol{y}) := \mathbf{V}(D_{\mathrm{ref}},\boldsymbol{y})$, i.e.

$$x = \boldsymbol{V}(\hat{x},\boldsymbol{y}),$$

we may now introduce the model problem transported to the reference domain which reads

$$\begin{aligned} -\operatorname{div}_{\hat{x}}\big(\boldsymbol{A}(\boldsymbol{y})\nabla_{\hat{x}}\hat{u}(\boldsymbol{y})\big) &= \hat{f}(\boldsymbol{y}) \quad \text{in } D_{\mathrm{ref}}, \\ \hat{u}(\boldsymbol{y}) &= 0 \qquad\ \ \text{on } \partial D_{\mathrm{ref}}. \end{aligned} \tag{6}$$

Herein, we have

$$\boldsymbol{A}(\hat{x},\boldsymbol{y}) := (\boldsymbol{J}^\intercal\boldsymbol{J})^{-1}(\hat{x},\boldsymbol{y}) \det \boldsymbol{J}(\hat{x},\boldsymbol{y}), \quad \hat{f}(\hat{x},\boldsymbol{y}) := (f\circ\boldsymbol{V})(\hat{x},\boldsymbol{y}) \det \boldsymbol{J}(\hat{x},\boldsymbol{y}) \tag{7}$$

and

$$\hat{u}(\hat{x}, \boldsymbol{y}) := (u \circ \boldsymbol{V})(\hat{x}, \boldsymbol{y}).$$

**Remark 2.2.** *The uniformity condition* (2) *implies that the functional determinant* $\det \mathbf{J}(\hat{x}, \boldsymbol{y})$ *in* (7) *is either uniformly positive or negative, see [16] for the details. We shall assume without loss of generality* $\det \boldsymbol{J}(\hat{x}, \boldsymbol{y}) > 0$ *and hence* $|\det \boldsymbol{J}(\hat{x}, \boldsymbol{y})| = \det \boldsymbol{J}(\hat{x}, \boldsymbol{y})$*, i.e. we my just drop the modulus. More precisely, due to* (2)*, we can bound the determinant according to*

$$0 < \frac{1}{C_{\text{uni}}^d} \leq \det \boldsymbol{J}(\hat{x}, \boldsymbol{y}) \leq C_{\text{uni}}^d < \infty$$

*for every* $\hat{x} \in D_{\text{ref}}$ *and almost every* $\boldsymbol{y} \in \Gamma$*. In addition, all singular values of* $\boldsymbol{J}^{-1}(\hat{x}, \boldsymbol{y})$ *are bounded from below by* $C_{\text{uni}}^{-1}$ *and from above by* $C_{\text{uni}}$*. From this, we obtain the bound*

$$0 < \frac{1}{C_{\text{uni}}^{d+2}} \leq \|\boldsymbol{A}(\hat{x}, \boldsymbol{y})\|_2 \leq C_{\text{uni}}^{d+2} < \infty \tag{8}$$

*for every* $\hat{x} \in D_{\text{ref}}$ *and almost every* $\boldsymbol{y} \in \Gamma$*. Hence, the transported model problem stays uniformly elliptic.*

We conclude this section by summarizing the regularity results for $\boldsymbol{A}, \hat{f}, \hat{u}$, cp. (6), with respect to the parameter $\boldsymbol{y} \in \Gamma$ from [16]. For this, denote by $\mathcal{F}$ the set of finitely supported multi-indices

$$\mathcal{F} := \{\mu \in \mathbb{N}_0^\infty ; \ |\operatorname{supp} \mu| < \infty\} \quad \text{where} \quad \operatorname{supp} \mu := \{m \in \mathbb{N} ; \ \mu_m \neq 0\}.$$

**Theorem 2.3.** *Let the right hand side* $f$ *from* (50) *satisfy* $\|\partial_x^{\boldsymbol{\alpha}} f\|_{L^\infty(\mathcal{D})} \leq c\boldsymbol{\alpha}!\rho^{-|\boldsymbol{\alpha}|}$ *for some constants* $c_f, \rho > 0$*. Then, it holds for every* $\boldsymbol{\alpha} \in \mathcal{F}$ *that*

$$\begin{aligned}
\left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\alpha}} \boldsymbol{A}\right\|_{L^\infty(D_{\text{ref}};\mathbb{R}^{d\times d})} &\leq C|\boldsymbol{\alpha}|!c^{|\boldsymbol{\alpha}|}\boldsymbol{\gamma}^{\boldsymbol{\alpha}}, \\
\left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\alpha}} \hat{f}\right\|_{L^\infty(D_{\text{ref}})} &\leq C|\boldsymbol{\alpha}|!c^{|\boldsymbol{\alpha}|}\boldsymbol{\gamma}^{\boldsymbol{\alpha}}, \\
\left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\alpha}} \hat{u}\right\|_{H^1(D_{\text{ref}})} &\leq C|\boldsymbol{\alpha}|!c^{|\boldsymbol{\alpha}|}\boldsymbol{\gamma}^{\boldsymbol{\alpha}},
\end{aligned}$$

*for some constants* $c, C$*, which depend on* $c_f, \rho, C_{\text{uni}}, d, D_{\text{ref}}, \|\boldsymbol{\gamma}\|_{\ell^1}$ *but are independent of the sequence* $\boldsymbol{\alpha}$*.*

## 3. Field discretisation

Let $\mathcal{T}_h$ denote a suitable triangulation of $D_{\text{ref}} \subset \mathbb{R}^d$ with mesh width $h > 0$. We introduce the finite element spaces

$$\mathcal{S}_h^1 := \left\{v \in C(\overline{D_{\text{ref}}}) : v|_T \in \Pi_1 \text{ for all } T \in \mathcal{T}_h\right\}$$

and

$$\mathcal{S}_h^0 := \left\{v \colon \overline{D_{\text{ref}}} \to \mathbb{R} : v|_T \in \Pi_0 \text{ for all } T \in \mathcal{T}_h\right\},$$

where $\Pi_1$ denotes the space of linear polynomials, while $\Pi_0$ is the space of constant polynomials.

In [20], it has been shown how the random vector field (5) can efficiently be represented by means of finite elements. This results in a representation

$$\boldsymbol{V}_h(\hat{x}, \boldsymbol{y}) = \hat{x} + \sum_{m=1}^M y_k \boldsymbol{V}_{m,h}(\hat{x}) = \hat{x} + \sum_{m=1}^M y_m \sum_{i=1}^d \sum_{k=1}^n c_{i,k,m} \varphi_k(\hat{x}) \boldsymbol{e}_i,$$

where $\boldsymbol{e}_1, \dots, \boldsymbol{e}_d$ denotes the canonical basis of $\mathbb{R}^d$, $\varphi_1, \dots \varphi_n$ is a basis for $\mathcal{S}_h^1$ and $c_{i,k,m} \in \mathbb{R}$ are the coefficients in the basis representation of $\boldsymbol{v}_{m,h}$. Consequently, we obtain

$$\boldsymbol{J}_h(\hat{x}, \boldsymbol{y}) = \boldsymbol{I} + \sum_{m=1}^{M} y_k \boldsymbol{v}'_{m,h}(\hat{x}) = \boldsymbol{I} + \sum_{m=1}^{M} y_k \sum_{i=1}^{d} \sum_{k=1}^{n} c_{i,k,m} \boldsymbol{e}_i \big(\nabla \varphi_k(\hat{x})\big)^{\mathsf{T}}.$$

More explicitly, the Jacobians $\boldsymbol{v}'_{m,h}(\hat{x})$ are given according to

$$\boldsymbol{V}'_{m,h}(\hat{x}) = \sum_{k=1}^{n} \begin{bmatrix} c_{1,k,m}\partial_1\varphi_k(\hat{x}) & \cdots & c_{1,k,m}\partial_d\varphi_k(\hat{x}) \\ \vdots & \ddots & \vdots \\ c_{d,k,m}\partial_1\varphi_k(\hat{x}) & \cdots & c_{d,k,m}\partial_d\varphi_k(\hat{x}) \end{bmatrix}$$

Since $\partial_i\varphi_k(\hat{x})$, $i = 1, \dots, d$, $k = 1, \dots, n$ are piecewise constant functions from $\mathcal{S}_h^0$, we can represent $\boldsymbol{v}'_{m,h}$ also in an element based fashion according to

$$\boldsymbol{V}'_{m,h} = \sum_{T \in \mathcal{T}_h} \begin{bmatrix} \tilde{c}_{T,m,1,1} & \cdots & \tilde{c}_{T,m,1,d} \\ \vdots & \ddots & \vdots \\ \tilde{c}_{T,m,d,1} & \cdots & \tilde{c}_{T,m,d,d} \end{bmatrix} \chi_T(\hat{x}) =: \sum_{T \in \mathcal{T}_h} \boldsymbol{C}_{T,m} \chi_T(\hat{x}),$$

where $\chi_T$ donotes the characteristic function of $T \in \mathcal{T}_h$ and $\tilde{c}_{T,m,i,j} \in \mathbb{R}$ are the corresponding coefficients. Hence, we end up with a piecewise constant representation of $\boldsymbol{V}'_h$, which reads

$$\boldsymbol{J}_h(\hat{x}, \boldsymbol{y}) = \boldsymbol{I} + \sum_{T \in \mathcal{T}_h} \left( \sum_{m=1}^{M} \boldsymbol{C}_{T,m} y_m \right) \chi_T(\hat{x}).$$

From this representation, it is straightforward to calculate $\det \boldsymbol{J}_h(\hat{x}, \boldsymbol{y})$ for a given $\boldsymbol{y} \in \Gamma$, also in an element based fashion. Having $\boldsymbol{V}_h(\hat{x}, \boldsymbol{y})$, $\boldsymbol{J}_h(\hat{x}, \boldsymbol{y})$, $\det \boldsymbol{J}_h(\hat{x}, \boldsymbol{y})$ at our disposal, it is then easy to evaluate $\boldsymbol{A}(\hat{x}, \boldsymbol{y})$ and $\hat{f}(\hat{x}, \boldsymbol{y})$, as well.

## 4. Adaptive Galerkin discretisation

In this section we describe the Galerkin discretization of the considered random PDE (6) in a finite dimensional subspace $\mathcal{V}_N \subset \mathcal{V} = \mathcal{X} \otimes \mathcal{Y} = H_0^1(D_{\mathrm{ref}}) \otimes L_\pi^2(\Gamma)$. The basis of $\mathcal{V}_N$ is given as tensor product of a first order Lagrange FE basis on a mesh representing $D_{\mathrm{ref}}$ and Legendre chaos polynomials orthonormal with respect to the joint probability density $\pi$ associated with $y$ in terms of Legendre polynomial chaos and finite elements (FE). Moreover, the residual based a posteriori error estimator of [3, 4] is recalled for the problem at hand.

For efficient computations of the Galerkin projection and the error estimator, the resulting system with inhomogeneous coefficient and right-hand side (7) is represented in the tensor train format as presented in Section 5.

4.1. **Parametric and deterministic discretization.** To determine a multivariate polynomial basis of $\mathcal{Y}$, we first define the full tensor index set of order $M \in \mathbb{N}$ by

$$\begin{aligned} \Lambda &:= \{(\mu_1, \dots, \mu_M, 0, \dots) \in \mathcal{F} : \mu_m = 0, \dots, d_m - 1,\ m = 1, \dots, M\} \\ &\simeq \Lambda_1 \times \ldots \times \Lambda_M \times \{0\} \ldots \subset \mathcal{F}, \end{aligned}$$

with complete index sets of size $d_m$ given by

$$\Lambda_m := \{0, \dots, d_m - 1\}, \quad m = 1, \dots, M.$$

For any such subset $\Lambda \subset \mathcal{F}$, we define $\operatorname{supp} \Lambda := \bigcup_{\mu \in \Lambda} \operatorname{supp} \mu \subset \mathbb{N}$. Let $(P_n)_{n=0}^{\infty}$ denote a basis of $L^2(\mathbb{R}, [-1, 1])$ orthonormal with respect to the Lebesgue measure consisting of Legendre polynomials $P_n$ of degree $n \in \mathbb{N}_0$ on $\mathbb{R}$. By tensorization of the univariate polynomials, an orthogonal basis of $L^2(\Gamma,[-1,1]^{\mathbb{N}})$ is obtained, see [23]. For any multi-index $\mu \in \mathcal{F}$, the tensor product polynomial $P_\mu := \bigotimes_{m=1}^\infty P_{\mu_m}$ in $y \in \Gamma$ is expressed as the finite product

$$P_\mu(y) = \prod_{m=1}^\infty P_{\mu_m}(y_m) = \prod_{m \in \operatorname{supp}\mu} P_{\mu_m}(y_m).$$

A discrete subspace of $\mathcal{X}$ is given by the conforming finite element space $\mathcal{X}(\mathcal{T}) := \operatorname{span}\{\varphi_i\}_{i=1}^N \subset \mathcal{X}$ of degree one on some simplicial regular mesh $\mathcal{T}$ of domain $D_{\mathrm{ref}}$ with the set of faces $\mathcal{S}$ (i.e., edges for $d=2$) and basis functions $\varphi_i$. Furthermore, we denote the piecewise constants on $\mathcal{T}$ by $\{\psi_i\}_{i=1}^{N_0}$. In order to circumvent complications due to an inexact approximation of boundary values, we assume that $D_{\mathrm{ref}}$ is a polygon. We denote by $P_1(\mathcal{T})$ the space of piecewise polynomials of degree one on the triangulation $\mathcal{T}$. The assumed first order FE discretization with Lagrange elements then satisfies $\mathcal{X}(\mathcal{T}) \subset P_1(\mathcal{T}) \cap C(\overline{\mathcal{T}})$. For any element $T \in \mathcal{T}$ and face $F \in \mathcal{S}$, we set the entity sizes $h_T := \operatorname{diam} T$ and $h_F := \operatorname{diam} F$. Let $n_F$ denote the exterior unit normal on any face $F$. The jump of some $\chi \in H^1(D_{\mathrm{ref}};\mathbb{R}^d)$ on $F = \overline{T_1} \cap \overline{T_2}$ in normal direction $[\![\chi]\!]_F$ is then defined by

$$[\![\chi]\!]_F := \chi|_{T_1} \cdot n_F - \chi|_{T_2} \cdot n_F. \tag{9}$$

By $\omega_T$ and $\omega_F$ we denote the element and facet patches defined by the union of all elements which share at least a vertex with $T$ or $F$, respectively. Consequently, the Clément interpolation operator $I\colon \mathcal{X} \to \mathcal{X}_p(\mathcal{T})$ satisfies, respectively for $T \in \mathcal{T}$ and $F \in \mathcal{S}$,

$$\|v - Iv\|_{L^2(T)} \le c_{\mathcal{T}} h_T |v|_{\mathcal{X},\omega_T}, \qquad \|v - Iv\|_{L^2(F)} \le c_{\mathcal{S}} h_F^{1/2} |v|_{\mathcal{X},\omega_F}, \tag{10}$$

where the seminorms $|\cdot|_{\mathcal{X},\omega_T}$ and $|\cdot|_{\mathcal{X},\omega_F}$ are the restrictions of $\|\cdot\|_{\mathcal{X}}$ to $\omega_T$ and $\omega_F$, respectively.

The fully discrete approximation space subject to some mesh $\mathcal{T}$ and active set $\Lambda$ with $|\Lambda| < \infty$ is given by

$$\mathcal{V}_N := \mathcal{V}_N(\Lambda;\mathcal{T},p) := \left\{ v_N(x,y) = \sum_{\mu\in\Lambda} v_{N,\mu}(x) P_\mu(y)\,;\;\; v_{N,\mu} \in \mathcal{X}_p(\mathcal{T}) \right\}, \tag{11}$$

and it holds $\mathcal{V}_N(\Lambda;\mathcal{T}) \subset \mathcal{V}$. We define a tensor product interpolation operator $\mathcal{I} : L^2(\Gamma,\pi;\mathcal{X}) \to \mathcal{V}_N(\Lambda;\mathcal{T},p)$ for $v = \sum_{\mu\in\mathcal{F}} v_\mu P_\mu \in L^2(\Gamma,\pi;\mathcal{X})$ by setting

$$\mathcal{I}v := \sum_{\mu\in\Lambda} (Iv_\mu) H_\mu. \tag{12}$$

For $v \in \mathcal{V}(\Lambda)$ and all $T \in \mathcal{T},\ F \in \mathcal{S}$, it holds

$$\|(\mathrm{id} - \mathcal{I})v\|_{L^2(\Gamma,\pi;L^2(T))} \le c_{\mathcal{T}} h_T |v|_{\mathcal{V},\omega_T}, \tag{13}$$

$$\|v - \mathcal{I}v\|_{L^2(\Gamma,\pi;L^2(F))} \le c_{\mathcal{S}} h_F^{1/2} |v|_{\mathcal{V},\omega_F}, \tag{14}$$

where

$$|v|^2_{\mathcal{V},\omega_T} := \int_\Gamma |v(y)|^2_{\mathcal{X},\omega_T}\,\mathrm{d}\pi(y), \qquad |v|^2_{\mathcal{V},\omega_F} := \int_\Gamma |v(y)|^2_{\mathcal{X},\omega_F}\,\mathrm{d}\pi(y).$$

4.2. **Variational formulation.** Using the transformation in (7), the weak formulation of the model problem (6) reads: find $u \in \mathcal{V}$, such that for all $v \in \mathcal{V}$ there holds

$$\int_\Gamma \int_{D_{\mathrm{ref}}} \boldsymbol{A}(\hat{x},\boldsymbol{y}) \nabla_{\hat{x}} u(\hat{x},\boldsymbol{y}) \cdot \nabla_{\hat{x}} v(\hat{x},\boldsymbol{y})\,\mathrm{d}\hat{x}\mathrm{d}\pi(\boldsymbol{y}) = \int_\Gamma \int_{D_{\mathrm{ref}}} f(\hat{x},\boldsymbol{y}) v(\hat{x},\boldsymbol{y})\,\mathrm{d}\hat{x}\mathrm{d}\pi(\boldsymbol{y}). \tag{15}$$

Employing the finite dimensional spaces of the previous section lead to the discrete weak problem: find $u \in \mathcal{V}_N$, such that for all $i,i' = 1,\dots,N$ and $\alpha,\alpha' \in \Lambda$:

$$\mathbf{L}(i,\alpha,i',\alpha') = F(i,\alpha).$$

Here, we define the discrete linear operator

$$\mathbf{L}(i,\alpha,i',\alpha') := \int_\Gamma \int_{D_{\mathrm{ref}}} \boldsymbol{A}(\hat{x},\boldsymbol{y}) \nabla_{\hat{x}} \varphi_i(\hat{x}) P_\alpha(\boldsymbol{y}) \nabla_{\hat{x}} \varphi_{i'}(\hat{x}) P_{\alpha'}(\boldsymbol{y}) \, \mathrm{d}\hat{x} \mathrm{d}\pi(\boldsymbol{y}) \tag{16}$$

and the discrete right-hand side

$$\mathbf{F}(i,\alpha) := \int_\Gamma \int_{D_{\mathrm{ref}}} f(\hat{x},\boldsymbol{y}) \varphi_i P_\alpha \mathrm{d}\pi \, \mathrm{d}\hat{x}.$$

4.3. **Residual based a posteriori error estimates.** In the following, we recall the residual based a posteriori error estimator derived in [3,4], adopted for the problem at hand. An efficient reformulation in the tensor train format is postponed to Section 5. The basis for the estimator is the residual $\mathcal{R}(w_N) \in L^2_\pi(\Gamma;\mathcal{X}^*) = \mathcal{V}^*$ with respect to some $w_N \in \mathcal{V}_N$ and the solution $u \in \mathcal{V}$ of (6) given by

$$\mathcal{R}(w_N) := \mathcal{A}(u - w_N) = f - \mathcal{A}(w_N).$$

It has an $L^2_\pi(\Gamma)$-convergent expansion in $(P_\nu)_{\nu\in\mathcal{F}}$ given by

$$\mathcal{R}(w_N) = \sum_{\nu\in\mathcal{F}} r_\nu(w_N) P_\nu,$$

with coefficients $r_\nu \in \mathcal{X}^*$ characterized by

$$\langle r_\nu, v\rangle = \int_{D_{\mathrm{ref}}} f_\nu v - \sum_{(\mu,\kappa)\in\Upsilon_\nu} \boldsymbol{A}_\mu \nabla w_{N,\kappa} \cdot \nabla v \, \mathrm{d}\hat{x} \qquad \forall v \in \mathcal{X}.$$

Here, $f_\nu$ and $\boldsymbol{A}_\nu$ denote the coefficients in the Legendre chaos expansion of $f = \sum_{\mu\in\mathcal{F}} f_\mu P_\mu$ and $A = \sum_{\kappa\in\mathcal{F}} A_\kappa P_\kappa$ and $\Upsilon_\nu := \{(\mu,\kappa) : \int_\Gamma P_\nu(y) P_\mu(y) P_\kappa(y) \, \mathrm{d}\pi(y) \neq 0\}$ is the $\nu$-relevant triple product tuple set.

We recall a central theorem from [3], which enables the derivation of an error bound based on an approximation $w_N$ of the Galerkin projection $u_N$ of the solution $u$ in the energy norm.

**Theorem 4.1.** *Let $\mathcal{V}_N \subset \mathcal{V}$ be a closed subspace and $w_N \in \mathcal{V}_N$, and let $u_N \in \mathcal{V}_N$ denote the $\mathcal{A}$ Galerkin projection of $u \in \mathcal{V}$ onto $\mathcal{V}_N$. Then, for some $c, c_{\mathcal{I}} > 0$, it holds*

$$\|u - w_N\|^2_{\mathcal{A}} \le \frac{1}{c} \left( \sup_{v\in\mathcal{V}} \frac{|\langle \mathcal{R}(w_N), (\mathrm{id} - \mathcal{I})v|}{\|v\|_{\mathcal{V}}} + c_{\mathcal{I}} \|u_N - w_N\|_{\mathcal{A}} \right)^2 + \|u_N - w_N\|^2_{\mathcal{A}}.$$

**Remark 4.2.** *We henceforth assume that the data $f$ and $A$ are exactly expanded in a finite set $\Delta$ with $\Lambda \subset \Delta \subset \mathcal{F}$, i.e. there is no significant contribution from the neglected modes $\mathcal{F} \setminus \Delta$. The residual can then be split into approximation and truncation contributions*

$$\mathcal{R}(w_N) = \mathcal{R}_\Lambda(w_N) + \mathcal{R}_{\Delta\setminus\Lambda}(w_N),$$

*where $\mathcal{R}_\Xi$ denotes the restriction of the expansion to the set $\Xi \subset \mathcal{F}$. Computable upper bounds for the two residual terms and the algebraic error $\|u_N - w_N\|_{\mathcal{A}}$ are recalled in the following.*

For any discrete $w_N \in \mathcal{V}_N$, we define the following error estimators in analogy to the presentation in [3,4] and [5]:

- A deterministic residual estimator for $\mathcal{R}_\Lambda$ steering the adaptivity of the mesh $\mathcal{T}$ is given by

$$\eta(w_N)^2 := \sum_{T\in\mathcal{T}} \eta_T(w_N)^2 + \sum_{S\in\mathcal{S}} \eta_S(w_N)^2, \tag{17}$$

with volume and facet contributions, for any $T \in \mathcal{T}$ and $S \in \mathcal{S}$,

$$\eta_T(w_N) := \|\sum_{\mu\in\Lambda}(f_\mu - \operatorname{div} \boldsymbol{A}_\mu \nabla w_{N,\mu})P_\mu\|_{L^2(\Gamma,\pi;L^2(T))}, \tag{18}$$

$$\eta_S(w_N) := \|\sum_{\mu\in\Lambda}[\![\boldsymbol{A}_\mu \nabla w_{N,\mu}]\!]P_\mu\|_{L^2(\Gamma,\pi;L^2(S))}. \tag{19}$$

- The stochastic truncation error estimator stems from the splitting of the residual (4.2), while considering the inactive part over $\mathcal{F}\setminus\Lambda$. It is possible to construct the estimator, as in the deterministic case, for every element of the triangulation and consider different mesh discretisations for every stochastic multi-index. Since we want to focus on a closed formulation and avoid technical details, the stochastic estimator is formulated on the whole domain $D_{\mathrm{ref}}$. For more insight, we introduce a collection of suitable tensor sets, which indicate the error portion of every active stochastic dimension $m = 1, \ldots, M$ (in fact, we could even consider $m > M$),

$$\Delta_m := \{\mu \in \mathcal{F} \mid \mu_j = 0, \ldots, d_j - 1,\ j = 1, \ldots, M, \tag{20}$$

$$j \neq m, \mu_m = d_m, \mu_k = 0, k > M\}. \tag{21}$$

Then, for every $w_N \in \mathcal{V}_N$, the stochastic tail estimator on $\Delta_m$ is given by

$$\zeta_m(w_N)^2 := \| \sum_{\mu\in\Delta_m} (f_\mu - \operatorname{div}(\boldsymbol{A}_\mu \nabla_{\hat{x}} w_{N,\mu}))P_\mu\|^2_{L^2(\Gamma,\pi;L^2(D_{\mathrm{ref}})}. \tag{22}$$

The collection of sets $\{\Delta_n\}_{n=1}^M$ is beneficial in the adaptive refinement procedure but it does not cover the whole stochastic contributions of the residual. For this, we need to compute the global stochastic tail estimator

$$\zeta(w_N)^2 := \sum_{\mu\in\mathcal{F}\setminus\Lambda} \zeta_\mu(w_N)^2, \tag{23}$$

which is an infinite sum that becomes finite due to orthogonality of the employed polynomials.

- The algebraic error, i.e. the distance of $w_N$ from the $\mathcal{V}_N$ best approximation $u_N$, in particular due to a finite termination of an iterative solver, can be bounded by

$$\|u_N - w_N\| \le \iota(w_N), \tag{24}$$

where

$$\iota(w_N) := \|(\boldsymbol{L}W - \boldsymbol{F})\boldsymbol{H}^{-1/2}\|_F.$$

Here, $W \in \mathbb{R}^{N,d_1,\ldots,d_M}$ denotes the coefficient tensor of $w_N \in \mathcal{V}_N$ and $\boldsymbol{L}$ is the discrete operator from (16). Note that the rank-1 operator $\boldsymbol{H}$ is a base change operator to orthonormalize the physical basis functions, i.e.,

$$\boldsymbol{H} := H_0 \otimes I \otimes \cdots \otimes I, \quad H_0(i,i') = \int_{D_{\mathrm{ref}}} \nabla_{\hat{x}}\varphi_i(\hat{x})\nabla_{\hat{x}}\varphi_{i'}(\hat{x})\mathrm{d}\hat{x}. \tag{25}$$

The combination of these estimators in the context of Theorem 4.1 yields an overall bound $\Theta$ for the energy error similar to the references [3, 4, 7] and [5]

**Corollary 4.3.** *For any $w_N \in \mathcal{V}_N$, the solution $u \in \mathcal{V}$ of the model problem* (50) *and the Galerkin approximation $u_N \in \mathcal{V}_N$ in* (15)*, there exists constants $c_\eta, c_\zeta, c_\iota > 0$ such that it holds*

$$\|w_N - u\|^2_{\mathcal{A}} \le \Theta := \big(c_\eta \eta(w_N) + c_\zeta \zeta(w_N) + c_\iota \iota(w_N)\big)^2 + \iota(w_N)^2. \tag{26}$$

**Remark 4.4.** *Observing the residual in* (4.3) *it becomes clear that the derived error estimators suffer from the "curse of dimensionality" and are hence not computable for larger problems. However, the hierarchical low-rank tensor representation introduced in the next section alleviates this substantial obstacle and makes possible the adaptive algorithms described in Section 6.*

## 5. Tensor trains

The inherent tensor structure of the involved Bochner function space $\mathcal{V} = \mathcal{X} \bigotimes_{m=1}^{M} \mathcal{Y}_m$ and the corresponding finite dimensional analogue $\mathcal{V}_N$ enables the use of tensor formats which aim at breaking the curse of dimensionality.

A typical representative $v \in \mathcal{V}_N$ can be written as

$$v(\hat{x}, \boldsymbol{y}) = \sum_{i=1}^{N} \sum_{\mu \in \Lambda} V(i,\mu) \varphi_i(\hat{x}) P_\mu(\boldsymbol{y}), \tag{27}$$

where $V \in \mathbb{R}^{N,d_1,\dots,d_M}$ is a high dimensional tensor containing for example the projection coefficients

$$V(i,\mu) = \mathbb{E}_\pi \left[ \int_{D_{\text{ref}}} v(\hat{x},\cdot) \varphi_i(\hat{x}) \mathrm{d}x P_\mu(\cdot) \right].$$

Here, setting $d = \max\{d_1,\dots,d_M\}$, the complexity of $V$ is $\mathcal{O}(Nd^M)$ and grows exponentially with the number of dimensions $M \in \mathbb{N}$ in the stochastic approximation space. To avoid this issue, we impose a *low-rank* assumption on the involved object and introduce the tensor train (TT) format as follows.

A tensor $V \in \mathbb{R}^{N,d_1,\dots,d_M}$ is called in *tensor train* format if every entry can be represented as the result of a matrix-vector multiplication:

$$V(i,\mu_1,\dots,\mu_M) = \sum_{k_0=1}^{r_0} \cdots \sum_{k_{M-1}=1}^{r_{M-1}} V_0(i,k_0) \prod_{m=1}^{M} V_m(k_{m-1},\mu_m,k_m). \tag{28}$$

To ease notation, one defines $r_M = 1$. If the vector $\mathbf{r} = (r_0,\dots,r_M)$ is minimal in some sense, we call $\mathbf{r}$ the *TT-rank* and (28) is the *TT*-decomposition of $V$. One observes, the complexity of $V$ can now be estimated linearly in the number of dimensions $\mathcal{O}(dM \max \mathbf{r}^2)$. In [19,22] it was shown that many functions in real wold applications admits a low-rank representation.

Given the full tensor description $V$ at hand, one can compute the tensor train representation by a *hierarchical singular value decomposition* (HSVD) [11]. Usually, this is unfeasible due to the high dimensionality of $V$ or since it is known only implicitly. In that case, the utilization of high-dimensional interpolation or regression algorithm is advisible [12,21].

We rely on the TT-reconstruction approach and us it for the representation of the transformed coefficient function and the right-hand-side (7).

### 5.1. Galerkin discretisation in tensor train format.

In the following we assume a tensor representations of the right-hand side and the coefficient function at hand. More precisely, we denote the low-rank approximations of (7) by

$$f^{\mathrm{TT}}(\hat{x}, \boldsymbol{y}) = \sum_{\mu \in \Lambda_f} \sum_{i=1}^{N_0} F(i,\mu) \psi_i(\hat{x}) P_\mu(\boldsymbol{y}), \tag{29}$$

where $F$ admits a TT representation of rank $\mathbf{r}^f$ and $\Lambda_f$ is a tensor multi-index set with local dimension cap $\mathbf{d}^f = (d_1^f,\dots,d_M^f)$. Analogously, every component of the symmetric matrix coefficient

$$\boldsymbol{A}(\hat{x}, \boldsymbol{y}) = \begin{bmatrix} \boldsymbol{a}_{0,0}(\hat{x},\boldsymbol{y}) & \boldsymbol{a}_{0,1}(\hat{x},\boldsymbol{y}) \\ \boldsymbol{a}_{1,0}(\hat{x},\boldsymbol{y}) & \boldsymbol{a}_{1,1}(\hat{x},\boldsymbol{y}) \end{bmatrix} \tag{30}$$

is approximated by $a_{i,j}^{\mathrm{TT}}$, $i,j \in \{0,1\}$ as in (29) with TT-ranks $\mathbf{r}^{i,j}$. Here, the order three component tensors of the approximated matrix entry $\boldsymbol{A}^{\mathrm{TT}}|_{i,j} = a_{i,j}^{\mathrm{TT}}$ are denoted by $\{a_{i,j}^m\}_{m=0}^M$.

**Remark 5.1.** *Since, for the coefficient, the TT reconstruction is carried out for every matrix entry in* (30)*, the local dimensions $d^{i,j} = (d_1^{i,j}, \dots, d_M^{i,j})$ and tensor ranks can vary among those four tensor trains. Here, we assume that every approximation has the same local dimensions and the tensor multi-index set covering those indices is denoted by $\Xi \subset \mathcal{F}$, possibly different from the solution active set $\Lambda$. As stated in [5], it is beneficial to chose $\Xi$ such that for all $\mu \in \Lambda$ also $2\mu = (2\mu_1, \dots, 2\mu_M) \in \Xi$. Due to the orthogonality structure of the polynomial basis, this feature guarantees a well-posed discrete problem and enables quasi-optimal convergence rates.*

**Remark 5.2.** *Usually, one defines the tensor multi-index set of the coefficient for a possibly larger expansion with $L > M$ modes. This gives the possibility to define a residual based error estimator, which provides an indication for the incorporation of additional dimensions in $\Lambda$. Since we focus on fixed finite dimensional noise in this article, we do not consider this more general setting. For more details, we refer to [3, 5, 7].*

On $\mathcal{V}_N$, the Galerkin operator resulting from the transformed weak problem in tensor train format is given as the sum of four TT operators such that for all $i, i' = 1, \dots, N$ and $\alpha, \alpha' \in \Lambda$

$$\mathbf{L}(i, \alpha, i', \alpha') = (\mathbf{L}_1 + \mathbf{L}_2 + \mathbf{L}_3 + \mathbf{L}_4)(i, \alpha, i', \alpha'), \tag{31}$$

each corresponding to one addend of the resulting matrix-vector product in (16).

We illustrate the explicit construction of the TT operator for the term $\mathbf{L}_1$. By denoting $\nabla^i g$ the $i$-th component of the gradient of a function $g$, one obtains for the first low-rank approximated bilinear form addend

$$\mathbf{L}_1(i, \alpha, i', \alpha') \approx \int_\Gamma \int_{D_{\mathrm{ref}}} a_{0,0}^{\mathrm{TT}}(\hat{x}, \boldsymbol{y}) \nabla^1 \varphi_i(\hat{x}) \nabla^1 \varphi_{i'}(\hat{x}) P_\alpha(\boldsymbol{y}) P_{\alpha'}(\boldsymbol{y}) \mathrm{d}\pi(\boldsymbol{y}) \mathrm{d}\hat{x}. \tag{32}$$

Using the multi-linear structure of $a_{0,0}^{\mathrm{TT}}$, one can decompose the operator addend as

$$\mathbf{L}_1(i, \alpha, i', \alpha') \approx \sum_{k_0, \dots, k_{M-1}}^{r_0^{0,0}, \dots, r_{M-1}^{0,0}} \mathbf{L}_0^1(i, i', k_0) \prod_{m=1}^{M} \mathbf{L}_m^1(k_{m-1}, \alpha, \alpha', k_m), \tag{33}$$

where the first component tensor $\mathbf{L}_0^1$ depends on the physical discretization only, i.e.,

$$\mathbf{L}_0^1(i, i', k_0) = \sum_{\ell=1}^{N_0} a_{0,0}^0(\ell, k_0) \int_{D_{\mathrm{ref}}} \langle \nabla^1 \varphi_i(\hat{x}), \psi_\ell(\hat{x}) \nabla^1 \varphi_{i'}(\hat{x}) \rangle \mathrm{d}\hat{x}. \tag{34}$$

The remaining tensor operator parts decompose into one dimensional integrals over an orthogonal polynomial triple product of the form

$$\mathbf{L}_m^1(k_{m-1}, \alpha, \alpha', k_m) = \sum_{\mu_m=0}^{d_m^{0,0}-1} a_{0,0}^m(k_{m-1} \mu_m, k_m) \int_{[-1,1]} P_{\mu_m} P_\alpha P_{\alpha'} \, \mathrm{d}\pi, \tag{35}$$

which is known explicitly thanks to the recursion formula for orthogonal polynomials [1, 8].

**Remark 5.3.** *Due to the sum of TT operators in* (31)*, the result can be represented by a tensor with TT-rank $4 \max\{\mathbf{r}^{i,j} \mid i, j \in \{0, 1\}\}$.*

By employing the tensor train approximation of $f_{\mathrm{TT}} \approx f$ and $\boldsymbol{a}_{i,j} \approx a_{i,j}^{\mathrm{TT}}$ we replace the original system of equations that need to be solved for $U \in \mathbb{R}^{N \times d_1 \times \ldots \times d_M}$

$$\mathbf{L} U = \mathbf{F} \tag{36}$$

with a constrained minimization problem on the low-rank manifold $\mathcal{M}_{\boldsymbol{r}}$ containing all tensor trains of dimensionality represented by $\Lambda$ and fixed rank $\boldsymbol{r}$

$$W^{\mathrm{TT}} = \operatorname{argmin}_{V\in\mathcal{M}_{\boldsymbol{r}}} \|\boldsymbol{L}^{\mathrm{TT}}V - \boldsymbol{F}^{\mathrm{TT}}\|_F. \tag{37}$$

Here, we take $\boldsymbol{L}^{\mathrm{TT}}$ and $\boldsymbol{F}^{\mathrm{TT}}$ as the tensor train approximations of $\boldsymbol{L}$ and $\boldsymbol{F}$, respectively and $\|\cdot\|_F$ as the Frobenius norm.

To solve (37), we chose a preconditioned alternating least square (ALS) algorithm as described in [7, 18].

This eventually results in an approximation of the true Galerkin solution of (15)

$$w_N := w(\Lambda,\mathcal{T},\boldsymbol{r},\tau) = \sum_{\mu\in\Lambda}\sum_{i=1}^{N} W^{\mathrm{TT}}(i,\mu)\varphi_i P_\mu. \tag{38}$$

## 6. Adaptive algorithm

The error estimator presented in Section 6.1 gives rise to an adaptive algorithm, which refines the spatial discretization, the stochastic polynomial set and the representation format iteratively based on local error estimators and indicators. This enables the assessment of the development of the actual (unknown) error. The inherently high computational complexity of the error estimators can be overcome by means of the tensor train formalism. In what follows, we examine the efficient computation of the individual error estimator components in the TT format and describe an adaptive algorithm. For more details and a more general framework, we refer to the presentations in [3, 4, 5].

6.1. **Efficient computation of error estimators.** We illustrate the efficient computation on the example of the deterministic error estimator. For each element $T \in \mathcal{T}$ of the triangulation, the error estimator is given by (18), which suffers from the curse of dimensionality, due to the sum over $\Lambda$. Employing the low-rank approximation $\boldsymbol{A}^{\mathrm{TT}} \approx \boldsymbol{A}$, $f^{\mathrm{TT}} \approx f$ and $w_N$ renders the computation feasible. To make this more explicit, we compute the norm

$$\eta_T(w_N)^2 = \|f^{\mathrm{TT}} - \operatorname{div}(\boldsymbol{A}^{\mathrm{TT}}\nabla_{\hat{x}} w_N)\|_{L^2(\Gamma,\pi;L^2(T))} \tag{39}$$

by equating the individual terms of the inner product

$$\eta_T(w_N)^2 = \|f^{\mathrm{TT}}\|_{L^2(\Gamma,\pi;L^2(T))} - 2\langle f^{\mathrm{TT}}, \operatorname{div}(\boldsymbol{A}^{\mathrm{TT}}\nabla_{\hat{x}} w_N)\rangle_{L^2(\Gamma,\pi;L^2(T))} \tag{40}$$

$$+ \|\operatorname{div}(\boldsymbol{A}^{\mathrm{TT}}\nabla_{\hat{x}} w_N)\|_{L^2(\Gamma,\pi;L^2(T))}. \tag{41}$$

The first term is a simple inner product of a functional tensor train, reduces to a simple summation over the tensor components due to the orthonormality of the polynomial basis, i.e.,

$$\|f^{\mathrm{TT}}\|_{L^2(\Gamma,\pi;L^2(T))} = \sum_{\mu\in\Lambda}\sum_{i'=1}^{N_0}\sum_{i=1}^{N_0} F(i,\mu)F(i',\mu)\int_T \psi_i\psi_{i'}\,\mathrm{d}\hat{x}, \tag{42}$$

whereas the high dimensional sum can be evaluated for every tensor dimension in parallel using, for all $i, i' = 1,\ldots,N_0$, that

$$\sum_{\mu\in\Lambda} F(i,\mu)F(i',\mu) = \sum_{k_0=1}^{r_0^f}\cdots\sum_{k_{M-1}=1}^{r_{M-1}^f}\sum_{k_0'=1}^{r_0^f}\cdots\sum_{k_{M-1}'=1}^{r_{M-1}^f} F_0(i,k_0)F_0(i,k_0') \tag{43}$$

$$\prod_{m=1}^{M}\sum_{\mu_m=1}^{d_m} F_m(k_{m-1},\mu_m,k_m)F_m(k_{m-1}',\mu_m,k_m'). \tag{44}$$

Note that the iterated sum over the tensor ranks has to be interpreted as matrix-vector multiplications. Hence, the formula above can be evaluated highly efficiently. In fact, if the employed TT format utilizes a component orthogonalization and $f^{\mathrm{TT}}$ is left-orthogonal, the product can be neglected and one only has to sum over $k_0$ and $k_0'$.

For the remaining terms in (40), one has to find a suitable representation of $\boldsymbol{A}^{\mathrm{TT}}\nabla_{\hat{x}}w_N$. Since taking the gradient is a linear operation, one can calculate a tensor representation of $\boldsymbol{A}^{\mathrm{TT}}w_N$ explicitly, involving multiplied ranks and doubled polynomial degrees. For a detailed construction, consider [5, Section 5]. The matrix-vector multiplication due to entry-wise TT representation of $\boldsymbol{A}^{\mathrm{TT}}$ does not impose any further difficulties but a slight increase in complexity since one needs to cope with a sum of invidiual portions.

Finally, the mixed and volume terms are computed in the same fashion, using the same arguments, as in (43).

6.2. **Fully adaptive algorithm.** Given an initial configuration consisting of a regular mesh $\mathcal{T}$, a finite tensor multi-index set $\Lambda \subset \mathcal{F}$, a (possibly random) start tensor $W^{\mathrm{TT}}$ with TT-rank $\mathbf{r}$ and solver parameter $\tau$, covering for example a termination threshold, rounding parameter, iteration limit or precision arguments, we now present the adaptive refinement process. The procedure is summarized in Algorithm 1.

Initially on every level, we decided to obtain the data approximation by a tensor reconstruction. The procedure is e.g. described in [6] and referred to as

$$f^{\mathrm{TT}},\ \boldsymbol{A}^{\mathrm{TT}} \leftarrow \mathrm{Reconstruct}[\Xi, \mathcal{T}, N_s], \tag{45}$$

where the multi-index set $\Xi$ can be chosen arbitrarily, but it is advisable to apply remark 5.1. The number of samples $N_s$ is used as a placeholder as well, since one is not limited to the use of random samples. Instead, it is beneficial to use sparse grids or adaptive quadrature rules. The numerical comparison is subject to further research and out of scope of this article. In what follows we assume that the approximations are sufficiently accurate.

The process of obtaining a numerical approximation $w_N \in \mathcal{V}_N$ is denoted by

$$w_N \leftarrow \mathrm{Solve}[\Lambda, \mathcal{T}, \tau, W^{\mathrm{TT}}]. \tag{46}$$

The used preconditioned ALS algorithm is only exemplary to obtain $w_N$. Alternative alternating methods or Riemannian algorithms are feasible as well.

One subsequently needs to compute the estimators. To obtain the overall estimator $\Theta(\eta, \zeta, \iota)$, one has to evaluate the individual components by the following methods

$$\begin{aligned}(\eta_T)_{T\in\mathcal{T}}, \eta &\leftarrow \mathrm{Estimate}_x[w_N, f^{\mathrm{TT}}, \boldsymbol{A}^{\mathrm{TT}}, \Lambda, \mathcal{T}, p]\\ (\zeta_m)_{m\in\mathbb{N}}, \zeta &\leftarrow \mathrm{Estimate}_{\boldsymbol{y}}[w_N, f^{\mathrm{TT}}, \boldsymbol{A}^{\mathrm{TT}}, \Lambda]\\ \iota &\leftarrow \mathrm{Estimate}_{\mathrm{LS}}[w_N, f^{\mathrm{TT}}, \boldsymbol{A}^{\mathrm{TT}}].\end{aligned}$$

A comparison of the global estimator values $\eta, \zeta$ and $\iota$ results in the refinement decision.

6.2.1. *deterministic refinement.* In case of a dominant deterministic error estimator $\eta$, one employs a Dörfler marking strategy on the mesh $\mathcal{T}$ for a ratio constant $\theta_\eta$. In abuse of notation, we use $(\eta_T)_{T\in\mathcal{T}}$ as the local error estimator on every triangle, where the jump components of $(\eta_S)_{S\in\mathcal{F}}$ are distributed among their nearby elements. The method, consisting of the marking process and the conforming refinement of the marked triangles is covered by

$$\mathcal{T} \leftarrow \mathrm{Refine}_x[(\eta_T)_{T\in\mathcal{T}}, \eta, \mathcal{T}, \theta_\eta]. \tag{47}$$

6.2.2. *stochastic refinement.* In case of the stochastic error estimator $\zeta$ exceeding the others, we apply a Dörfler marking on the set of local estimators $(\zeta_m)_{m\in\mathbb{N}}$ until the prescribed ratio $0 < \theta_\zeta < 1$ is reached. Note that in the finite dimensional noise case, we have $\zeta_m = 0$ for $m > M$. Afterwards, the marked dimensions in $\Lambda$ are increased $d_m \leftarrow d_m + 1$ by the method

$$\Lambda \leftarrow \mathrm{Refine}_{\boldsymbol{y}}[(\zeta_m)_{m\in\mathbb{N}}, \zeta, \Lambda, \theta_\zeta]. \tag{48}$$

**Remark 6.1.** *As stated in Section 4.3 the global estimator $\zeta$ is not the sum of the individual estimators $(\zeta_m)_{m\in\mathbb{N}}$, since the coupling structure is more involved. Hence, we use $\zeta_{\mathrm{sum}} := \sum_{m\in\mathbb{N}} \zeta_m$ in the marking procedure. Due to the regularity of the solution (Theorem 2.3), for $\Lambda$ large enough, one has $\zeta_{\mathrm{sum}} \approx \zeta$.*

6.2.3. *algebraic refinement.* In the end, if $\iota$ has the largest contribution to the error, we “refine the solver”. For simplicity, we fix most of the solver parameter such as the number of alternating iteration or the termination value to low values that can be seen as overcautious. Nevertheless, in the low-rank tensor framework, the model class is restricted by the TT-rank $\mathbf{r}$. Hence, we then allow $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{1}$ and add a random rank 1 tensor onto the solution tensor $W^{\mathrm{TT}}$. We summarize this methodology in

$$W^{\mathrm{TT}}, \tau \leftarrow \mathrm{Refine}_{\mathrm{LS}}[W^{\mathrm{TT}}, \tau]. \tag{49}$$

6.2.4. *adaptive stochastic Galerkin method.* One global iteration of this algorithm refines either the deterministic mesh $\mathcal{T}$, the stochastic polynomial index-set $\Lambda$ or the tensor rank $\mathbf{r}$. Repetition until the defined estimator $\Theta(\eta, \zeta, \iota)$ in Corollary 4.3 falls below a desired accuracy $\epsilon > 0$, yields the adaptively constructed low-rank approximation $w_N \in \mathcal{V}_N$.

**Algorithm 1:** Reconstruction based adaptive stochastic Galerkin method

```
input : Initial guess $w_N$ with solution coefficient $W^{\mathrm{TT}}$;
        solver accuracy $\tau$;
        mesh $\mathcal{T}$ with degrees $p$;
        index set $\Lambda$;
        Sample size for reconstruction $N_s$;
        Dörfler marking parameter $\theta_\eta$ and $\theta_\zeta$;
        desired accuracy $\epsilon$.
output: New solution $w_N$ with new solution coefficient $W^+$;
        new mesh $\mathcal{T}^+$, or new index set $\Lambda^+$, or new tolerance $\tau^+$.

repeat
    $f^{\mathrm{TT}}, \boldsymbol{A}^{\mathrm{TT}}$ ← $\mathrm{Reconstruct}[\Xi, \mathcal{T}, N_s]$
    $w_N$ ← $\mathrm{Solve}[\Lambda, \mathcal{T}, \tau, W^{\mathrm{TT}}]$
    $(\eta_T)_{T\in\mathcal{T}}, \eta$ ← $\mathrm{Estimate}_x[w_N, f^{\mathrm{TT}}, \boldsymbol{A}^{\mathrm{TT}}, \Lambda, \mathcal{T}, p]$
    $(\zeta_m)_{m\in\mathbb{N}}, \zeta$ ← $\mathrm{Estimate}_{\boldsymbol{y}}[w_N, f^{\mathrm{TT}}, \boldsymbol{A}^{\mathrm{TT}}, \Lambda]$
    $\iota$ ← $\mathrm{Estimate}_{\mathrm{LS}}[w_N, f^{\mathrm{TT}}, \boldsymbol{A}^{\mathrm{TT}}]$

    switch $\max\{\eta, \zeta, \iota\}$ do
        case $\eta$ do $\mathcal{T}$ ← $\mathrm{Refine}_x[(\eta_T)_{T\in\mathcal{T}}, \eta, \mathcal{T}, \theta_\eta]$;
        case $\zeta$ do $\Lambda$ ← $\mathrm{Refine}_{\boldsymbol{y}}[(\zeta_m)_{m\in\mathbb{N}}, \zeta, \Lambda, \theta_\zeta]$;
        case $\iota$ do $W^{\mathrm{TT}}, \tau$ ← $\mathrm{Refine}_{\mathrm{LS}}[W^{\mathrm{TT}}, \tau]$;
until $\Theta(\eta, \zeta, \iota) < \epsilon$;
return $w_N^+ = w_N$; $\mathcal{T}^+ = \mathcal{T}$; $\Lambda^+ = \Lambda$; $\tau^+ = \tau$
```

| Circle | | L-shape | |
|---|---|---|---|
| tolerance $\hat{\epsilon}$ | KL-terms $M$ | tolerance $\hat{\epsilon}$ | KL-terms $M$ |
| 0.5 | 5 | 0.5 | 6 |
| 0.1 | 21 | 0.1 | 21 |

FIGURE 1. Comparison of different truncation values and their implication on the number of KL modes for the employed reference domains.

## 7. NUMERICAL EXAMPLES

This section is concerned with the demonstration of the performance of the adaptive algorithm depicted in the preceding section.

We consider the model problem with a constant right-hand side

$$\begin{aligned} -\operatorname{div}\big(\nabla u(\omega)\big) &= 1 \quad \text{in } D(\omega), \\ u(\omega) &= 0 \quad \text{on } \partial D(\omega) \end{aligned} \tag{50}$$

on two different reference domains in $\mathbb{R}^2$, namely the unit-circle and the L-shape. The Karhunen-Loève expansion stems from a Gaussian covariance kernel

$$\operatorname{Cov}[\boldsymbol{V}](\hat{x},\hat{x}') = \frac{1}{1000}\begin{bmatrix} 5\exp(-2\|\hat{x}-\hat{x}'\|_2^2) & \exp(-0.1\|2\hat{x}-\hat{x}'\|_2^2) \\ \exp(-0.1\|\hat{x}-2\hat{x}'\|_2^2) & 5\exp(-0.5\|\hat{x}-\hat{x}'\|_2^2) \end{bmatrix}. \tag{51}$$

The random variables in the Karhunen-Loève expansion are assumed to be independent and uniformly distributed on $[-\sqrt{3},\sqrt{3}]$, i.e. they have normalized variance. Moreover, the mean is given by the identity, i.e. $\mathbb{E}[\boldsymbol{V}](\hat{x}) = \hat{x}$.

The computed spectral decomposition is truncated at a given threshold $\hat{\epsilon}$, which takes different values in the computational examples. In Table 7 we summarize how the choice of the truncation parameter influences the number of involved stochastic dimensions.

Our main focus is on the correct approximation of the solution mean

$$\mathbb{E}\left[u(\hat{x},\cdot)\right] = \int_\Gamma u(\hat{x},\boldsymbol{y})\mathrm{d}\pi(\boldsymbol{y}) \tag{52}$$

by means of the adaptive low-rank Galerkin approximation. Therefore, all experiments involve the computation of a reference mean, based on Monte Carlo sampling. To that end, we employ the anisotropic sparse grid quadrature with Gauss-Legendre points[1], as described in [13]. The mean is then calculated on a fine reference mesh, resulting from uniform refinement of the last, adaptively computed, mesh, having at least $10^5$ degrees of freedom. The number of quadrature points is choosen such that at least $1000$ nodes are involved. We denote this object as $\mathbb{E}_{\mathrm{ref}}[u]$, where $u \in \mathcal{V}$ is a obtained sampling-wise.

**Remark 7.1.** *In the low-rank tensor train format, the mean of a function, given in orthonormal polynomials, is computed highly efficiently, since the set of employed polynomials is orthonormal w.r.t. the constant function. Since, the corresponding coefficient is already incorporated in the representation, computing the mean boils down to a simple tensor evaluation. More precisely, given* $u \in \mathcal{V}_N$ *we*

[1] The implementation can be found online: `https://github.com/muchip/SPQR`

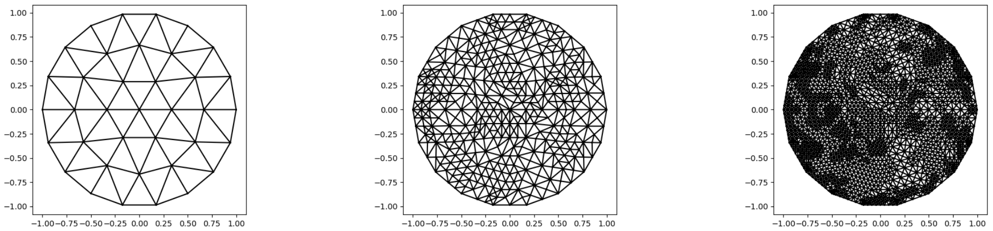

FIGURE 2. Adaptively refined mesh according to deterministic error estimator with mesh refinement ratio $\theta_\eta = 0.2$ and KL truncation tolerance $\hat{\epsilon} = 0.1$.

*compute*

$$\mathbb{E}[u(\hat{x},\cdot)] = \sum_{i=1}^{N}\sum_{\mu\in\Lambda} U(i,\mu)\varphi_i(\hat{x})\mathbb{E}[P_\mu(\cdot)] = \sum_{i=1}^{N} U^{\mathrm{TT}}(i,\mathbf{0})\varphi_i(\hat{x}) \tag{53}$$

*and evaluating the tensor train $U^{\mathrm{TT}}$ at the multi-index $\mathbf{0} = (0,\dots,0)$ consists of $M$-matrix-vector multiplications.*

The considered quantity of interest is the error of the mean to the reference. Therefore, for any TT approximation $w_N \in \mathcal{V}_N$ we compute, using remark 7.1

$$\|e\| := \|\mathbb{E}_{\mathrm{ref}}[u] - \mathbb{E}[w_N]\|_{H_0^1(D_{\mathrm{ref}})}\|\mathbb{E}_{\mathrm{ref}}[u]\|_{H_0^1(D_{\mathrm{ref}})}^{-1}. \tag{54}$$

Regarding the tensor reconstruction algorithm, we utilize the open-source library *xerus* [24]. Every approximated tensor is constructed on a set of $1000$ random samples $y \in \Gamma$ and polynomial degrees that are determined by the solution approximation. To be more precise, having a solution tensor $U^{\mathrm{TT}}$ of dimensionality $\mathbb{R}^{N,d_1,\dots,d_M}$, we employ the same stochastic dimensionality $d_1,\dots,d_M$ for the rhs $f^{\mathrm{TT}}$ and twice the amount $2d_1,\dots,2d_M$ for every coefficient matrix entry. In Section 3 is stated, that the coefficient and rhs are computed on pice-wise constant finite element functions, hence the deterministic dimension is here equal to $N_0$, instead of $N$, as for the solution. For the solution of the partial differential equation we make use of the PDE library *FEniCS* [9].

7.1. **Example 1.** The first considered example is the random domain problem, where we consider the reference domain to be the unit-circle. We use this problem as a reference, since the adaptive refinement yields similar results to uniform mesh refinement. Some occurring mesh refinements are given in 7.1.

For illustration purposes, we include the computational mean of the solution on the unit-disc, together with realisations of the transformed reference domain in Figure 4 and Figure 3.

Numerical test involving the adaptive refinement of the mesh are given in Figure 5. The initial configuration is set to $54$ cells and fixed polynomial degree in the stochastic space of $d_1 = \ldots = d_M = 1$ and tensor rank $\mathbf{r} = \mathbf{1}$, We observe that the considered relative mean error behaves just as the estimator commands.

7.2. **Example 2.** The second example is the L-shape $[-1,1]^2 \setminus \{[0,1]\times[-1,0]\}$. The corner singularity is a typical example where adaptive refinement yields better approximation rates w.r.t. degrees of freedom, in contrast to uniform refinement.

Starting with an initial configuration of $24$ cells and fixed polynomial degree in the stochastic space of $d_1 = \ldots = d_M = 1$ and tensor rank $\mathbf{r} = \mathbf{1}$, the described adaptive Galerkin algorithm, using the deterministic error estimator yields an adaptively refined mesh, displayed in Figure 7.2.

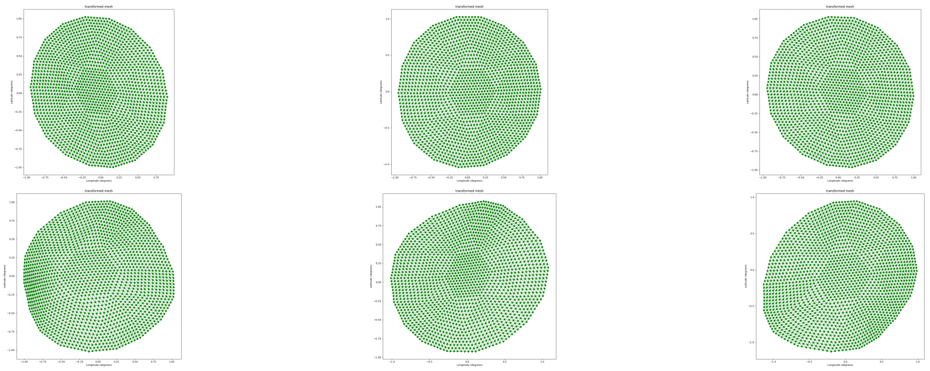

FIGURE 3. We show sample realisations of the transformed unit-disc with tolerance $\hat{\epsilon} = 0.5$ (top-row) and $\hat{\epsilon} = 0.1$ (bottom-row).

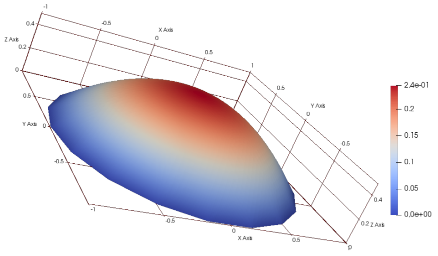

FIGURE 4. Computational mean of the unit-circle problem for tolerance $\hat{\epsilon} = 0.1$.

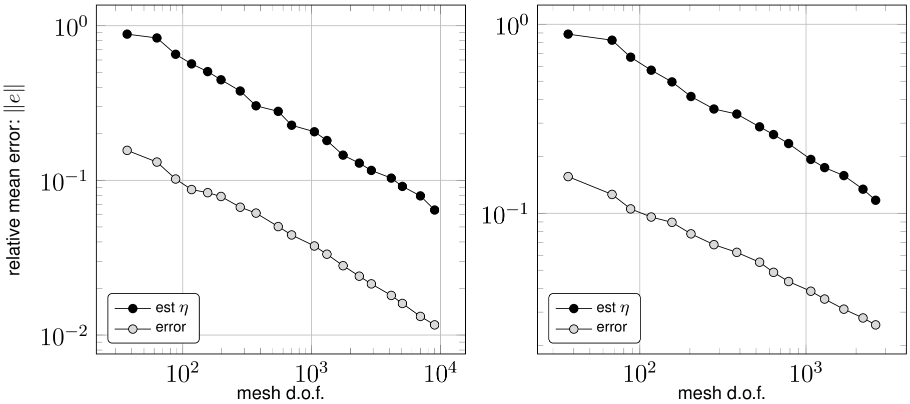

FIGURE 5. Deterministic estimator and error of the mean approximation with respect to a reference mean on the unit-circle. The tolerance is set to $\hat{\epsilon} = 0.5$ (left) and $\hat{\epsilon} = 0.1$ (right). The error follows the estimator.

For illustration, we highlight the approximated mean and a collection of realisations of the transformed reference domain in Figure 9 and Figure 8.

Finally, numerical comparison of the mean, as described in (54) is given in Figure 7.

## REFERENCES

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Applied mathematics series. Dover Publications, N. Chemsford, MA, 1964.

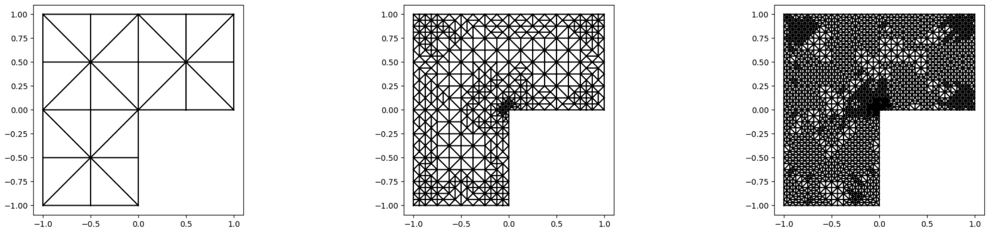

FIGURE 6. Adaptively refined mesh according to deterministic error estimator with mesh refinement ratio $\theta_\eta = 0.5$ and KL truncation tolerance $\hat{\epsilon} = 0.5$.

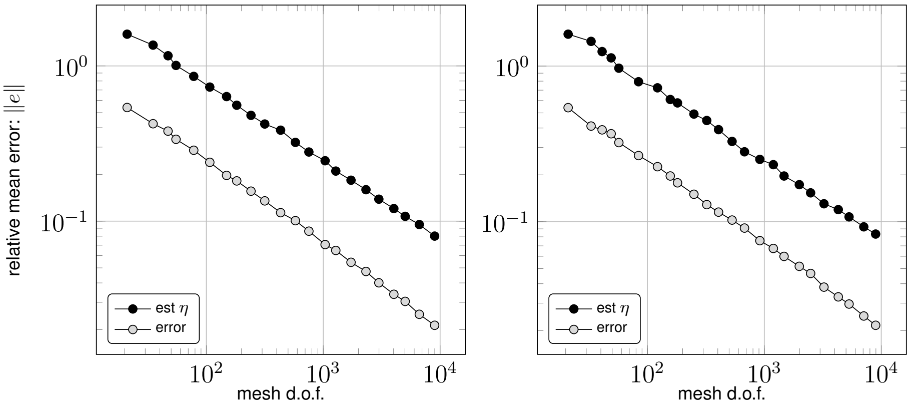

FIGURE 7. Deterministic estimator and error of the mean approximation with respect to a reference mean on the L-shape. The tolerance is set to $\hat{\epsilon} = 0.5$ (left) and $\hat{\epsilon} = 0.1$ (right). The error follows the estimator.

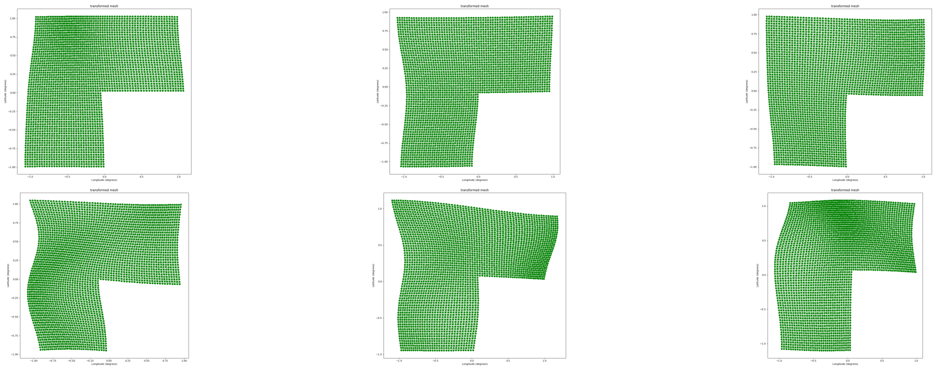

FIGURE 8. We show sample realisations of the transformed L-shape with tolerance $\hat{\epsilon} = 0.5$ (top-row) and $\hat{\epsilon} = 0.1$ (bottom-row).

[2] J. E. Castrillon-Candas, F. Nobile, and R. F. Tempone. Analytic regularity and collocation approximation for elliptic {PDEs} with random domain deformations. *Computers & Mathematics with Applications*, 71(6):1173–1197, 2016.

[3] Martin Eigel, Claude Jeffrey Gittelson, Christoph Schwab, and Elmar Zander. Adaptive stochastic Galerkin FEM. *Comput. Methods Appl. Mech. Engrg.*, 270:247–269, 2014.

[4] Martin Eigel, Claude Jeffrey Gittelson, Christoph Schwab, and Elmar Zander. A convergent adaptive stochastic Galerkin finite element method with quasi-optimal spatial meshes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(5):1367–1398, 2015.

[5] Martin Eigel, Manuel Marschall, Max Pfeffer, and Reinhold Schneider. Adaptive stochastic galerkin fem for lognormal coefficients in hierarchical tensor representations. *arXiv preprint arXiv:1811.00319*, 2018.

[6] Martin Eigel, Johannes Neumann, Reinhold Schneider, and Sebastian Wolf. Non-intrusive tensor reconstruction for high-dimensional random pdes. *Computational Methods in Applied Mathematics*.

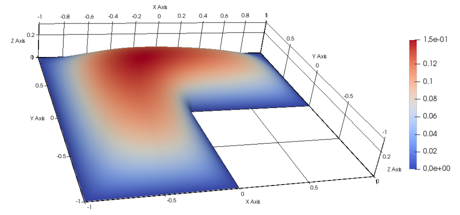

FIGURE 9. Computational mean of the L-shape problem for tolerance $\hat{\epsilon} = 0.1$.

[7] Martin Eigel, Max Pfeffer, and Reinhold Schneider. Adaptive stochastic Galerkin FEM with hierarchical tensor representations. *Numerische Mathematik*, 136(3):765–803, Jul 2017.

[8] Oliver G Ernst, Antje Mugler, Hans-Jörg Starkloff, and Elisabeth Ullmann. On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(2):317–339, 2012.

[9] FEniCS Project - Automated solution of Differential Equations by the Finite Element Method, `http://fenicsproject.org`.

[10] R. N. Gantner and M. D. Peters. Higher order quasi-monte carlo for baysian shape inversion. *SIAM/ASA J. Uncertain. Quantif.*, 6(2):707–736, 2018.

[11] Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[12] Lars Grasedyck, Melanie Kluge, and Sebastian KraÌĹmer. Variants of alternating least squares tensor completion in the tensor train format. *SIAM Journal on Scientific Computing*, 37(5):A2424–A2450, 2015.

[13] A.-L. Haji-Ali, H. Harbrecht, M. D. Peters, and M. Siebenmorgen. Novel results for the anisotropic sparse grid quadrature. *Journal of Complexity*, 47:62–85, 2018.

[14] H. Harbrecht, M. Peters, and R. Schneider. On the low-rank approximation by the pivoted Cholesky decomposition. *Applied numerical mathematics*, 62:28–440, 2012.

[15] H. Harbrecht, M. Peters, and M. Siebenmorgen. Efficient approximation of random fields for numerical applications. *Numerical Linear Algebra with Applications*, 22(4):596–617, 2015.

[16] H. Harbrecht, M. Peters, and M. Siebenmorgen. Analysis of the domain mapping method for elliptic diffusion problems on random domains. *Numerische Mathematik*, 134(4):823–856, 2016.

[17] E. Hille and R. S. Phillips. *Functional analysis and semi-groups*, volume 31. American Mathematical Society, Providence, 1957.

[18] Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM Journal on Scientific Computing*, 34(2):A683–A713, 2012.

[19] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[20] M. D. Multerer. A note on the domain mapping method with rough diffusion coefficients. *arXiv:1805.02889*, 2018.

[21] Ivan Oseledets and Eugene Tyrtyshnikov. Tt-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010.

[22] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

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

[24] General purpose library for numerical calculations with higher order tensors., `http://libxerus.org`.

[25] D. Xiu and D. M. Tartakovsky. Numerical methods for differential equations in random domains. *SIAM Journal on Scientific Computing*, 28(3):1167–1185, 2006.