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

Preprint ISSN 2198-5855

Adaptive stochastic Galerkin FEM with hierarchical tensor representations

Martin Eigel[1], Max Pfeffer[2], Reinhold Schneider[2]

submitted: September 8, 2015

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

[2] Technische Universität Berlin
Institut für Mathematik
Straße des 17. Juni 136
10623 Berlin
Germany
E-Mail: pfeffer@math.tu-berlin.de
schneidr@math.tu-berlin.de

No. 2153
Berlin 2015

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, operator equations, 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/

ABSTRACT. The solution of PDE with stochastic data commonly leads to very high-dimensional algebraic problems, e.g. when multiplicative noise is present. The Stochastic Galerkin FEM considered in this paper then suffers from the curse of dimensionality. This is directly related to the number of random variables required for an adequate representation of the random fields included in the PDE. With the presented new approach, we circumvent this major complexity obstacle by combining two highly efficient model reduction strategies, namely a modern low-rank tensor representation in the tensor train format of the problem and a refinement algorithm on the basis of a posteriori error estimates to adaptively adjust the different employed discretizations. The adaptive adjustment includes the refinement of the FE mesh based on a residual estimator, the problem-adapted stochastic discretization in anisotropic Legendre Wiener chaos and the successive increase of the tensor rank. Computable a posteriori error estimators are derived for all error terms emanating from the discretizations and the iterative solution with a preconditioned ALS scheme of the problem. Strikingly, it is possible to exploit the tensor structure of the problem to evaluate all error terms very efficiently. A set of benchmark problems illustrates the performance of the adaptive algorithm with higher-order FE. Moreover, the influence of the tensor rank on the approximation quality is investigated.

1. INTRODUCTION

In the present paper we consider an adaptive stochastic Galerkin method for a class of PDE with random coefficients. In this respect, the presented approach is a continuation of the papers [20, 21, 22] of one of the authors. Therein, a PDE with stochastic data was reformulated as a parametric PDE in the Cameron Martin space, depending on infinitely many parameters. Our main contribution is the use of modern *hierarchical tensor representations* and *tensor product approximations* for the discretization and solution representation by which the *curse of dimensionality* is circumvented. We thus tackle a major obstacle for the employment of (intrusive) Galerkin methods with stochastic problems which usually lead to huge coupled algebraic systems that may quickly become prohibitive to setup and solve. The efficient low-rank tensor representation is combined with an adaptive scheme. This relies on (residual-based) reliable a posteriori error estimators. It is noteworthy that the evaluation of the a posteriori error estimator terms can be carried out very efficiently by exploiting the low-rank tensor format. The combination of these model reduction techniques results in a fully adaptive stochastic Galerkin scheme with computable overall error bounds for the mean square error in the energy norm. We remark that the a posteriori error estimation and tensor techniques could in principle also be applied to numerical solutions obtained by other numerical methods such as Stochastic Collocation (SC), see e.g. [35] and [7] for a similar approach in an adaptive Galerkin scheme.

Stochastic Galerkin FEM (SGFEM) have become one of the main numerical methods for stochastic PDE since their introduction in [32]. This framework was taken up again for a more thorough analysis beginning with [15, 29, 49, 2, 5, 4] and succeeding works. In contrast to sampling methods such as Monte-Carlo, SGFEM are based on an orthogonal projection onto a discrete subspace of the infinite dimensional stochastic problem space. A basis for this space is given by polynomials orthogonal with respect to the probability distribution of the random variables determining the model. This is due to the famous theorem of Cameron and Martin, see also [24, 67] for the extension to generalized polynomial chaos and recent convergence results for the representation in suitable Wiener chaos polynomials. Depending on the considered problem, the larger effort of this approach with respect to implementation and computation is rewarded by significantly higher convergence rates. Moreover, the best approximation property

(perturbed by inexact algebraic solvers) allows for the application of well-known deterministic error estimates in all active modes and a separation of deterministic and stochastic error contributions. This can be considered an advantage over stochastic interpolation methods such as the Stochastic Collocation (see [3, 51, 50, 66]), in particular with respect to the a priori stability of the approximation. Nevertheless, several of the derived concepts could probably be transferred. For an in-depth review of current numerical methods and the analysis of parametric PDE as in our context, we refer to [61, 35]. The topic of a posteriori error control for deterministic FEM is a fairly mature area and the interested reader is referred to [1, 65, 8] and [9] for a unified view on residual-based error estimation as used here.

We consider elliptic differential operators which depend affine linearly on a countable set of parameters $(y_m)_{m=1}^{\infty}$ with $y_m \in [-1,1]$, cf. [61]. Thus, the PDE can be viewed as a parametric PDE depending on $y \in \Gamma := [-1,1]^{\infty}$. This type of dependence with respect to countable random variables is for instance obtained by the common Karhunen-Loève (KL) expansion for the stochastic data. Due to the decaying singular values of the KL decomposition of the stochastic conductivity field, we obtain a canonical ordering of stochastic variables in the Legendre chaos polynomials. Subsequently, we take advantage of the natural tensor structure of the problem by employing the simple tensor train (TT) format [53, 40].

The accuracy of the stochastic Galerkin approximation is determined by a set of parameters which affect: (a) the FE grid and the order of the FE basis, (b) the order of the individual chaos polynomials and (c) the truncation of the used KL expansion of the coefficient field. Moreover, as a new aspect in comparison to previous works, (d) the approximation by the low-rank tensor representation has to be taken into account. All parameters determining the approximation quality have to be chosen such that these error contributions are equilibrated.

In this paper, we derive a posteriori error estimators which measure all these contributions to the total error individually, i.e., the approximation error of the FEM discretization, the truncation error of the affine coefficient expansion, the stochastic discretization error subject to the degree of chaos polynomials and the approximation error of the low-rank tensor product representation. The error estimators for the first part were developed in [20] and are transferred to the tensor setting. Here, it is crucial to avoid the prohibitive complexity typically induced by the full discrete stochastic tensor space which can now be used instead of a small selection of stochastic polynomials as in earlier works. In fact, the numerical approximation with this hugely larger approximation space only becomes feasible due to the employed tensor compression. With the presented fully adaptive algorithm, we refine the FEM grid, enlarge the (anisotropic) active set of chaos polynomials and increase the multi-linear tensor approximation according to the corresponding a posteriori error estimators. With this approach, we try to exploit the sparsity of the elliptic model problem (2.1) and discover an “optimal” discretization setting with minimal effort. Results regarding the sparsity of the solution discretized in polynomial chaos for the class of PDE at hand can e.g. be found in [13, 12]. These insights are crucial to explain why adaptive methods for the infinite dimensional parametric problem may actually work very effectively. In principle, the present strategy can be employed with any existing adaptive finite element code without requiring further modifications, i.e. non-intrusively. Moreover, other equations of the problem class can be treated similarly as was shown in [20].

For the Galerkin method of the deterministic FE part, we use a classic residual-based approach although more recent techniques could be considered. However, it should be emphasized that we *do not require Galerkin orthogonality* for the FEM solution in the derivation of reliable error

bounds. In particular, the theory allows for inexact solutions. Hence, our main result can as well be applied to approximate solutions which are determined by means of other numerical methods.

For the present setting with affine noise, we make use of the fact that the polynomial degree in the stochastic discretization can increase by at most one in any stochastic dimension. For the more general case where the parameter dependence of the operator is approximated by a low-order chaos polynomial (e.g. in the lognormal setting), the respective residual could be computed in the same way as presented. However, this would include higher polynomial degrees which leads to more coupling terms and a higher computational complexity. Nevertheless, we expect that the described tensor representation would still permit an efficient computation of the error terms in such a generalized setting and we will devote some future research to this topic.

The described a posteriori error estimators require summations over all coefficients from the active chaos polynomials. It is a key aspect of this work that we can circumvent this prohibitively high complexity by applying the tensor product concepts to these summations. As a result, we obtain a significantly better scaling behavior than previous algorithms in [20, 21, 22]. The error of the tensor product approximation can be computed efficiently in the employed tensor format in an ℓ^2 sense equivalent to the present energy norm topology. This is due to the fact that the underlying energy norm in the tensor product space $\mathcal{V} := H^1(D) \otimes L^2(\Omega)$ is a cross-norm which can be transferred easily to a discrete ℓ^2 setting. This observation also provides a preconditioning which is crucial for defining an appropriate metric on the tensor subspace.

One crucial point is the choice of an appropriate tensor format. Tensor product approximation in the canonical format,

$$U(x_1, \dots, x_d) = \sum_{k=1}^{r_c} U_1(x_1, k) U_2(x_2, k) \cdots U_d(x_d, k),$$

also known as CANDECOMP, PARAFAC etc. [46], is one of the oldest approaches in applied mathematics where it is also known as *separation of variables*. Among others, it is dating back to Fourier. The canonical format can be seen as the generalization of the singular value decomposition (SVD) to higher-order tensors. The SVD was introduced to matrices and general linear operators by Schmidt [59]. While this generalization is quite intuitive and has therefore been studied extensively, it has severe drawbacks and is still not fully understood, see for example [47]. In this work, we pursue the alternative approach of regarding the SVD as a subspace approximation, e.g. POD (proper orthogonal decomposition), which leads to the concept of Tucker approximation [48] or the improved hierarchical Tucker approximation [39]. We focus on the special case of the modern TT representation [54] and the relation to the notion of reduced basis functions [10, 11]. Note however, that the resulting parameterization of a tensor is not really new, as it has been used in different fields of science, e.g. in quantum physics, where it is known as matrix product states (MPS). In the same context, general hierarchical tensors are called tree tensor networks, see [62] for an extensive review of the applications in physics. The hierarchical tensor representation, and hence tensor trains in particular, have many substantial features which render them highly attractive for computational purposes [40] and have lead to increased interest in the community in recent years. Since we cannot provide a detailed description of hierarchical tensor representations, we refer to the monograph [36], the initializing papers [53, 34, 39] and review articles [40, 38].

Tensor product approximation in the canonical format has been applied in a similar context in [44]. The TT format has been used in [43] for the lognormal case of the random diffusion problem, see

also [18, 27]. The canonical and the hierarchical formats have been investigated in [26]. In the lognormal case, a further approximation of the operator is required. To avoid this, nonintrusive Stochastic Collocation (SC) instead of intrusive Stochastic Galerkin methods have been used. For this, similar to (quasi-)Monte-Carlo methods, one computes the tensor $u(x, y)$ at several points $y_m, m = 1, \dots, M_{\mathrm{MC}}$, and employs a tensor completion procedure in order to find the tensor with TT rank r that fits these measurements the best. For an appropriately and adaptively choosen set of points, known as *adaptive cross approximation*, a tensor which fits the values exactly can be determined. Although the design of the set of points is somewhat heuristical, this approach does indeed work well, as confirmed by numerical examples [55, 16, 6]. However, we follow a different approach in this paper.

Tensors of fixed multi-linear rank form an analytic Riemannian manifold and allow for the application of related local optimization methods. For the sake of simplicity, we use an alternating least squares (ALS) method, which can be understood as an alternating block coordinate search or a nonlinear Gauß-Seidel method. Note that any other (more advanced) tensor algorithm for the solution of the linear system could be applied equally well, but the simple ALS can be used to better illustrate some features of the chosen format, which might be important for (and transferred to) other methods. Convergence of the solution with increasing rank has been demonstrated for instance in [44, 17]. The speed of convergence is mainly determined by the decay of the singular values in the KL decomposition. From this perspective, it seems that the tensor approach is less well suited for weak correlation. However, in the present case, high regularity of the true solution is only required for very fine FE grids, which are adaptively chosen. So far, we have not considered the overall complexity and convergence analysis, as has been done e.g in [21]. This will be part of a forthcoming paper.

The paper is organized as follows. Section 2 is concerned with the formulation of the problem setting and the stochastic Galerkin discretization in Legendre chaos polynomials. We then provide a brief introduction to reduced basis functions and the hierarchical tensor representation of the model in the TT format in Section 3. Additionally, the manifold structure of TT tensors is discussed. Section 4 introduces the ALS algorithm and goes into some detail about preconditioning and rank adaptivity. The different a posteriori error estimators are introduced in Section 5. We demonstrate how the tensor representation can be used to efficiently compute the required quantities while avoiding the curse of dimensionality. These error estimators are then combined in a fully adaptive algorithm which is described in Section 6. In the final Section 7, we present numerical examples which illustrate the performance of the developed adaptive algorithm based on different decay rates of the coefficients in the expansion of the coefficient. Moreover, we investigate the effect the TT rank has on the approximation quality of the numerical solution.

2. Setting and Discretization

In the following, we introduce the considered problem formally, present its weak formulation and describe the employed discretizations in finite dimensional function spaces.

2.1. **Model problem.**

We consider the elliptic second order linear model problem

$$-\operatorname{div}(a\nabla u) = f \quad \text{in } D, \qquad u|_{\partial D} = 0 \tag{2.1}$$

on a bounded Lipschitz domain $D \subset \mathbb{R}^d$ with $d = 1, 2, 3$ and assume for the sake of simplicity that the source term $f \in L^2(D)$ is deterministic. The diffusion coefficient a is a random field

on the probability space (Ω, Σ, P) over $L^\infty(D)$, see e.g. [61, 12]. For (2.1) to have a unique solution, we assume uniform ellipticity which is obtained if there exist constants $0 < a_{\min} \leq a_{\max} < \infty$ such that

$$a_{\min} \leq a(x,\omega) \leq a_{\max} \qquad \text{for all } (x,\omega) \in D \times \Omega. \tag{2.2}$$

With the Lax-Milgram lemma it immediately follows the realization-wise existence of a solution $u(\cdot,\omega) \in \mathcal{X} := H_0^1(D)$ for every $\omega \in \Omega$ which satisfies the variational formulation of the problem,

$$\int_D a(x,\omega)\nabla u(x,\omega)\cdot\nabla v(x,\omega)\,\mathrm{d}x = \int_D f(x)v(x)\,\mathrm{d}x \qquad \text{for all } v \in \mathcal{X}. \tag{2.3}$$

For $v \in \mathcal{X}$, we define the norm $\|v\|_{\mathcal{X}}^2 = (v,v)_{\mathcal{X}} := \int_D |\nabla v|^2\,\mathrm{d}x$ and denote the dual space of $\mathcal{X}$ by $\mathcal{X}^* = H^{-1}(D)$. The solution $u = u(x,\omega)$ is a random field in the probability space (Ω, Σ, P) and it holds

$$\sup_{\omega\in\Omega}\|u(\cdot,\omega)\|_{\mathcal{X}} \leq a_{\min}^{-1}\|f\|_{\mathcal{X}^*}. \tag{2.4}$$

Details regarding the regularity of u can be found in [13, 61, 12].

2.2. **Weak formulation.** We consider parametric elliptic problems of the form (2.1) which depend on a countable infinite set of parameters $y := (y_m)_{m=1}^\infty \in \Gamma := \bigtimes_{m=1}^\infty \Gamma_m$. The solution is in the tensor product Hilbert space

$$\mathcal{V} := \mathcal{X} \otimes \mathcal{Y} \quad \text{with} \quad \mathcal{X} := H_0^1(D) \quad \text{and} \quad \mathcal{Y} := \bigotimes_{m=1}^\infty L^2_{\pi_m}(\Gamma_m) \tag{2.5}$$

where π_m is the measure associated with y_m. Equation (2.1) then is equivalent to

$$\mathcal{A}(y)u(x,y) = f(x) \quad \text{in } D, \qquad u(x,y) = 0 \quad \text{on } \partial D \tag{2.6}$$

where the operator $\mathcal{A}(y) : H_0^1(D) \to H^{-1}(D)$ is given by

$$v \mapsto -\operatorname{div}(a(x,y)\nabla v) \quad \text{for } y \in \Gamma, \tag{2.7}$$

$f \in L^2(D)$ and $u \in \mathcal{V}$. The coefficient is assumed to admit a representation of the form

$$a(x,y) := a_0 + \sum_{m=1}^\infty a_m(x)y_m \quad \text{with } a_m \in L^\infty(D),\ m > 0, \tag{2.8}$$

which could e.g. be a Karhunen-Loève expansion. The operator (2.7) can then be expanded as

$$\mathcal{A}(y) = A_0 + \sum_{m=1}^\infty A_m y_m \qquad \text{for all } y \in \Gamma \tag{2.9}$$

where

$$A_m : H_0^1(D) \to H^{-1}(D), \quad v \mapsto -\operatorname{div}(a_m\nabla v). \tag{2.10}$$

Moreover, to ensure strong ellipticity of $\mathcal{A}$, we require

$$|\sum_{m=1}^\infty a_m(x)y_m| \leq \gamma a_0 \qquad \text{with} \qquad \gamma < 1 \tag{2.11}$$

which implies $a(x,y) \geq (1-\gamma) > 0$ and $|y_m| \leq 1$. Hence, $y \in \Gamma := [-1,1]^\infty$.

In what follows, we assume the parameters y_m to be independent and distributed identically and uniformly with probability measure $\pi_m = \pi_1$ for $m \in \mathbb{N}$. The product measure $\pi = \bigotimes_{m=1}^{\infty} \pi_1$ then is the probability measure of y on Γ. Thus, we now work in the image space $L^2_\pi(\Gamma)$ instead of $L^2_P(\Omega)$. Note that the identical distribution of the parameters is not a requirement but mainly simplifies the notation.

We obtain the weak formulation of (2.6) by integration with respect to π, i.e.,

$$\int_\Gamma \langle \mathcal{A}(y)u(y), v(y)\rangle \,\mathrm{d}\pi(y) = \int_\Gamma \int_D f(x)v(x,y)\,\mathrm{d}x\,\mathrm{d}\pi(y) =: \ell(v), \tag{2.12}$$

where $\langle\cdot,\cdot\rangle$ denotes the usual duality pairing in vector spaces. The left-hand side of (2.12) is a scalar product on $\mathcal{V} := L^2_\pi(\Gamma;\mathcal{X})$ for $w, v \in \mathcal{V}$,

$$(w,v)_{\mathcal{A}} := \int_\Gamma \langle \mathcal{A}(y)w(y), v(y)\rangle \,\mathrm{d}\pi(y) = \int_\Gamma \int_D a(x,y)\nabla w(x,y)\cdot\nabla v(x,y)\,\mathrm{d}x\,\mathrm{d}\pi(y) \tag{2.13}$$

which induces the energy norm $\|v\|^2_{\mathcal{A}} := (v,v)_{\mathcal{A}}$ for $v \in \mathcal{V}$. Moreover, we define

$$(w,v)_{A_0} := \int_\Gamma \langle A_0 w(y), v(y)\rangle \,\mathrm{d}\pi(y) = \int_\Gamma \int_D a_0(x)\nabla w(x,y)\cdot\nabla v(x,y)\,\mathrm{d}x\,\mathrm{d}\pi(y) \tag{2.14}$$

and the induced mean energy norm $\|v\|^2_{A_0} := (v,v)_{A_0}$. It is crucial that this is a *cross norm*, i.e. for elementary $w \in \mathcal{V},\ w(x,y) = w_x(x)w_y(y)$ it holds

$$\|w\|^2_{A_0} = \int_D a_0(x)\nabla w_x(x)\cdot\nabla w_x(x)\,\mathrm{d}x \int_\Gamma w_y(y)w_y(y)\,\mathrm{d}\pi(y) = (a_0 w_x, w_x)_{\mathcal{X}}(w_y,w_y)_{\mathcal{Y}}. \tag{2.15}$$

where

$$(u,v)_{\mathcal{X}} := \int_D \nabla u\cdot\nabla v\,\mathrm{d}x \quad \text{and} \quad (w,z)_{\mathcal{Y}} := \int_\Gamma wz\,\mathrm{d}\pi(y).$$

Note that this does not hold for the energy norm $\|w\|_{\mathcal{A}}$. However, we have equivalence of these two norms, $\|w\|_{\mathcal{A}} \sim \|w\|_{A_0}$, see e.g. [33]. The cross norm allows us to exploit the tensor structure of the solution and evaluate the norms independently, see [36]. Furthermore, after the discretization, we obtain equivalence to an ℓ^2 setting in which we can efficiently compute the error norms. Existence and uniqueness of the solution u of (2.12) follow from the Riesz representation theorem and u coincides with the solution of (2.1) for π-a.e. $y \in \Gamma$.

Note that for the numerical solution of the problem, the dimension of the parameter space is restricted to some $M < \infty$ such that a set of parameters $y_M \in \Gamma_M := [-1,1]^M$ has finite cardinality and

$$a_M(x,y) := a_0 + \sum_{m=1}^{M} a_m(x)y_m. \tag{2.16}$$

Then, the solution u_M of (2.6) is in the space

$$\mathcal{V}_M := \mathcal{X}\otimes\mathcal{Y}_M \quad \text{with} \quad \mathcal{Y}_M := \bigotimes_{m=1}^{M} L^2_{\pi_1}([-1,1]). \tag{2.17}$$

Hence, the truncation parameter M determines the dimension of the stochastic space in the discretization. An adaptive determination of M is part of the presented refinement algorithm.

For the discretization of the weak problem (2.12) with the stochastic Galerkin FEM, an orthonormal system of polynomials as a basis for the stochastic space is defined. Upon a selection of a (finite) active set Λ of stochastic modes, this leads to a semi-discretization of the problem and a semi-discrete best-approximation u_Λ of u. The fully discrete system is obtained with the discretization of the deterministic space by higher-order FEM. By this, the fully discrete Galerkin approximation u_N is obtained. Let $\mathcal{F}$ be the set of finitely supported multi-indices

$$\mathcal{F} \coloneqq \{\mu \in \mathbb{N}_0^\infty \mid |\operatorname{supp}\mu| < \infty\} \tag{2.18}$$

where $\operatorname{supp}\mu \coloneqq \{m \in \mathbb{N} \mid \mu_m \neq 0\}$ and $|\mu| \coloneqq \sum_{i\in\operatorname{supp}\mu} \mu_i$. For any subset $\Lambda \subset \mathcal{F}$, we define $\operatorname{supp}\Lambda \coloneqq \bigcup_{\mu\in\Lambda} \operatorname{supp}\mu \subset \mathbb{N}$. The infinite set

$$\partial\Lambda \coloneqq \{\nu \in \mathcal{F} \setminus \Lambda \mid \exists m \in \mathbb{N} : \nu - \epsilon_m \in \Lambda \vee \nu + \epsilon_m \in \Lambda\} \tag{2.19}$$

defines the boundary of Λ. Likewise, the active boundary of Λ is defined by

$$\partial^\circ\Lambda \coloneqq \{\nu \in \mathcal{F} \setminus \Lambda \mid \exists m \in \operatorname{supp}\Lambda : \nu - \epsilon_m \in \Lambda \vee \nu + \epsilon_m \in \Lambda\} \tag{2.20}$$

which is a finite set in case that $|\Lambda| < \infty$. Here, $\epsilon_m \coloneqq (\delta_{mn})_{n=1}^\infty$ denotes the Kronecker sequence.

Tensor product orthogonal polynomial basis. Let $(P_n)_{n=0}^\infty$ denote an orthogonal polynomial basis of $L^2_{\pi_1}([-1,1])$ of degree $\deg(P_n) = n$, see [61]. Due to the symmetry of π_1, such a basis satisfies a recursion of the form

$$\beta_n P_n(y_m) = y_m P_{n-1}(y_m) - \beta_{n-1} P_{n-2}(y_m) \quad \text{for } n \geq 1 \tag{2.21}$$

with $P_0 \coloneqq 1$ and $\beta_0 \coloneqq 0$. Since we assume a uniform distribution $\mathrm{d}\pi_1(y_m) = \frac{1}{2}\,\mathrm{d}y_m$, the $(P_n)_{n=0}^\infty$ are Legendre polynomials and $\beta_n = (4 - n^{-2})^{-1/2}$.

An orthogonal basis of $L^2_\pi(\Gamma)$ is obtained by tensorization of the univariate polynomials. For any $\mu \in \mathcal{F}$, the tensor product polynomial $P_\mu \coloneqq \bigotimes_{m=1}^\infty P_{\mu_m}$ in $y \in \Gamma$ is expressed as the finite product

$$P_\mu(y) = \prod_{m=1}^\infty P_{\mu_m}(y_m) = \prod_{m\in\operatorname{supp}\mu} P_{\mu_m}(y_m). \tag{2.22}$$

Recursion (2.21) implies

$$y_m P_\mu(y) = \beta_{\mu_m+1} P_{\mu+\epsilon_m}(y) + \beta_{\mu_m} P_{\mu-\epsilon_m}(y). \tag{2.23}$$

Moreover, we set $P_\mu \coloneqq 0$ for $\mu_m < 0$. The family of polynomials $(P_\mu)_{\mu\in\mathcal{F}}$ forms an orthonormal basis of $L^2_\pi(\Gamma)$, see [61].

Stochastic discretization. The solution u of (2.12) in the basis $(P_\mu)_{\mu\in\mathcal{F}}$ of $L^2_\pi(\Gamma)$ with coefficients $u_\mu \in \mathcal{X}$ for $\mu \in \mathcal{F}$ has the $L^2_\pi(\Gamma;\mathcal{X})$ convergent expansion

$$u(x,y) = \sum_{\mu\in\mathcal{F}} u_\mu(x) P_\mu(y). \tag{2.24}$$

The sequence of coefficients $(u_\mu)_{\mu\in\mathcal{F}} \in \ell^2(\mathcal{F};V)$ is determined by the infinite coupled system

$$A_0 u_\mu + \sum_{m=1}^\infty A_m \left(\beta_{\mu_m+1} u_{\mu+\epsilon_m} + \beta_{\mu_m} u_{\mu-\epsilon_m}\right) = f\delta_{\mu 0} \quad \text{for } \mu \in \mathcal{F}. \tag{2.25}$$

In contrast to preceding publications such as [20], we allow (and in fact require by construction) the set Λ to be a full tensor set with dimensions d_m

$$\Lambda = \{(\mu_1, \dots, \mu_M, 0, \dots) \in \mathcal{F} \mid \mu_m = 0, \dots, d_m - 1;\ m = 1, \dots, M\}. \tag{2.26}$$

This yields a set of $\prod_{m=1}^{M} d_m$ elements and for $d_m > 1$, the cardinality grows exponentially with M. While this results in a significant enlargement of the problem as compared to other publications where Λ was only assumed to be monotone (also called *downward closed*), it also yield a much more accurate solution. It is crucial that we break this *curse of dimensionality* in the following by means of low-rank tensor techniques.

For any subset $\Lambda \subset \mathcal{F}$, we define the (stochastically) semi-discrete space

$$\mathcal{V}(\Lambda) := \left\{ v_\Lambda(x, y) = \sum_{\mu \in \mathcal{F}} v_{\Lambda,\mu}(x) P_\mu(y) \mid v_{\Lambda,\mu} \in \mathcal{X}\ \forall\ \mu \in \Lambda \right\} \subset \mathcal{V}. \tag{2.27}$$

Furthermore, let $\mathcal{Y}_m := \operatorname{span}\{P_{\mu_m} : \mu_m = 0, \dots, d_m - 1\}$. Since Λ is a full tensor set,

$$\mathcal{V}(\Lambda) = \mathcal{X} \otimes \mathcal{Y}_M(\Lambda) := \mathcal{X} \otimes \left(\bigotimes_{m \in \operatorname{supp} \Lambda} \mathcal{Y}_m \right). \tag{2.28}$$

The Galerkin projection of u onto $\mathcal{V}(\Lambda)$ is the unique $u_\Lambda \in \mathcal{V}(\Lambda)$ which satisfies

$$(u_\Lambda, v)_{\mathcal{A}} = \ell(v) \quad \text{for all } v \in \mathcal{V}(\Lambda). \tag{2.29}$$

If Λ is finite, the sequence of coefficients $(u_{\Lambda,\mu})_{\mu \in \mathcal{F}} \in \mathcal{X}^\Lambda := \prod_{\mu \in \mathcal{F}} \mathcal{X}$ of u_Λ is determined by the finite system

$$A_0 u_{\Lambda,\mu} + \sum_{m=1}^{\infty} A_m \left(\beta_{\mu_m + 1} u_{\Lambda, \mu + \epsilon_m} + \beta_{\mu_m} u_{\Lambda, \mu - \epsilon_m} \right) = f \delta_{\mu 0} \quad \text{for } \mu \in \Lambda \tag{2.30}$$

where we set $u_{\Lambda,\nu} = 0$ for $\nu \in \mathcal{F} \setminus \Lambda$. Note that all terms in the sum (2.30) vanish for $m \in \mathbb{N} \setminus \operatorname{supp} \Lambda$.

2.3. **Deterministic discretization.** We discretize the deterministic space $\mathcal{X}$ by a conforming finite element space $\mathcal{X}_p(\mathcal{T}) \subset \mathcal{X}$ spanned by piecewise polynomials of degree p on some simplicial triangulation $\mathcal{T}$ of D. The nodal basis is given by $\{\varphi_i\}_{i=0}^{N} - 1$ with $N := \dim V_p(\mathcal{T})$. For the sake of a simple presentation, we assume D to be a polygon which is partitioned by $\mathcal{T}$. The set of edges (sides) is denoted $\mathcal{S}$ and $\mathcal{S} \cap D$ is the set of interior edges, $\mathcal{S} \cap \partial D$ the set of boundary edges. Likewise, for any $T \in \mathcal{T}$, the set $\mathcal{S} \cap \partial T$ contains the edges of $\mathcal{T}$ in the boundary of T. Moreover, for any $T \in \mathcal{T}$ and $S \in \mathcal{S}$, let $h_T := \operatorname{diam} T$ and $h_S := \operatorname{diam} S$ denote the element and edge sizes and let ω_T and ω_S define the patches of T and S, i.e., the union of all elements of $\mathcal{T}$ sharing at least a vertex with T or S, respectively.

The fully discrete space for the approximation of the solution is defined by

$$\mathcal{V}_p(\Lambda; \mathcal{T}) := \left\{ v_N(x, y) = \sum_{\mu \in \mathcal{F}} v_{N,\mu}(x) P_\mu(y) \mid \forall\ \mu \in \Lambda,\ v_{N,\mu} \in \mathcal{X}_p(\mathcal{T}) \right\} \subset \mathcal{V}(\Lambda). \tag{2.31}$$

Similar to (2.29), the Galerkin projection of u is the unique $u_N \in \mathcal{V}_p(\Lambda; \mathcal{T})$ which satisfies

$$(u_N, v)_{\mathcal{A}} = \ell(v) \quad \text{for all } v \in \mathcal{V}_p(\Lambda; \mathcal{T}). \tag{2.32}$$

The sequence of coefficients $(u_{N,\mu})_{\mu\in\Lambda} \in \mathcal{X}_p(\mathcal{T})^\Lambda = \prod_{\mu\in\Lambda} \mathcal{X}_p(\mathcal{T})$ for $\mu \in \Lambda$ is determined by

(2.33)
$$\int_D (A_0 u_{N,\mu}) v \,\mathrm{d}x + \sum_{m=1}^{\infty} \int_D A_m \left(\beta_{N,\mu_m+1} u_{\mu+\epsilon_m} + \beta_{\mu_m} u_{N,\mu-\epsilon_m}\right) v \,\mathrm{d}x = \int_D f \delta_{\mu 0} v \,\mathrm{d}x$$

for all $v \in \mathcal{X}_p(\mathcal{T})$. Note that we set $u_{N,\nu} = 0$ for $\nu \in \mathcal{F} \setminus \Lambda$ as before.

2.4. **Tensor structure.** The tensor structure of $\mathcal{V}$ (2.5) and its discretization $\mathcal{V}_p(\Lambda;\mathcal{T})$ (2.31) leads to a natural formulation of (2.32) in tensor notation. According to (2.31), we may decompose any $u_N \in \mathcal{V}_p(\Lambda;\mathcal{T})$ by

(2.34)
$$u_N(x,y) = \sum_{i=0}^{N-1} \sum_{\mu\in\Lambda} U(i,\mu)\varphi_i(x) P_\mu(y).$$

With the truncation parameter M in (2.16), we obtain the parameter tensor $U \in \mathbb{R}^{N\times d_1\times\cdots\times d_M}$. Set

(2.35)
$$K_m(i,j) := \int_D a_m(x) \nabla\varphi_i(x) \cdot \nabla\varphi_j(x) \,\mathrm{d}x, \qquad i,j = 0,\ldots,N-1,$$

(2.36)
$$B_m(\mu,\nu) := \int_{\Gamma_m} y_m P_{\mu_m}(y_m) P_{\nu_m}(y_m) \,\mathrm{d}\pi(y_m), \qquad \nu,\mu \in \mathcal{F},$$

for $m = 0,\ldots,M$, and with $y_0 \equiv 1$. Then, problem (2.32) can be rewritten as

(2.37)
$$\mathbf{A}(U) := \left(\sum_{m=0}^{M} \mathbf{A}_m\right)(U) = F$$

with

(2.38)
$$\mathbf{A}_m := K_m \otimes I \otimes \ldots \otimes B_m \otimes \ldots \otimes I,$$

(2.39)
$$F := f \otimes e_1 \otimes \ldots \otimes e_1,$$

where U is in tensor format as detailed in Section 3.2 and e_1 denotes the first unit vector.

3. Tensor Decomposition Formats

Since the set Λ grows exponentially with M, calculating the related sum in (2.34) becomes unfeasible very quickly for larger M. As a remedy for this problem, model reduction or compression techniques such as *reduced basis functions* and the *Tensor Train format* can be employed. We introduce these intimately related notions in the following sections.

3.1. **Reduced Basis Functions.** We seek to find a reduced basis $\{\Phi^{(0)}_{k_1} : k_1 = 1,\ldots,r_1\}$ that is given by a linear combination

(3.1)
$$\Phi^{(0)}_{k_1}(x) = \sum_{i=0}^{N-1} U_0(i,k_1)\varphi_i(x).$$

These functions span the subspace $\mathcal{L}_0 := \operatorname{span}\{\Phi_{k_1}^{(0)} : k_1 = 1, \dots, r_1\} \subseteq \mathcal{X}_p(\mathcal{T}) \subset \mathcal{X}$. This representation is in general not exact but instead is a further approximation of U as in (2.34) via a truncated SVD of

$$U(i, \mu_1, \dots, \mu_M) \approx \sum_{k_1=1}^{r_1} U_0(i, k_1) U_{\geq 1}(k_1, \mu_1, \dots, \mu_M). \tag{3.2}$$

For $u_N \in \mathcal{V}_p(\Gamma; \mathcal{T})$, we hence obtain

$$u_N(x, y) = \sum_{k_1=1}^{r_1} \sum_{\mu \in \Lambda} U_{\geq 1}(k_1, \mu) \Phi_{k_1}^{(0)}(x) P_\mu(y). \tag{3.3}$$

The crucial point is that it yields a sufficiently good representation *for all parameters* $\mu \in \Lambda$. This can be extended to all other modes

$$\mathcal{L}_m := \operatorname{span}\{\Phi_{k_{m+1}}^{(m)} : k_{m+1} = 1, \dots, r_{m+1}\} \subset \mathcal{X}_p(\mathcal{T}) \otimes \mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m \tag{3.4}$$

via

$$\begin{aligned} &\Phi_{k_{m+1}}^{(m)}(x, y_1, \dots, y_m) \\ &\quad = \sum_{i=0}^{N-1} \sum_{\mu_1=0}^{d_1-1} \cdots \sum_{\mu_m=0}^{d_m-1} U_{\leq m}(i, \mu_1, \dots, \mu_m, k_{m+1}) \varphi_i(x) P_{\mu_1}(y_1) \cdots P_{\mu_m}(y_m). \end{aligned} \tag{3.5}$$

These reduced basis sets do not circumvent the curse of dimensionality by themselves. However, we observe that the subspaces are nested, namely

$$\mathcal{L}_{m+1} \subset \mathcal{L}_m \otimes \mathcal{Y}_{m+1}. \tag{3.6}$$

This nestedness implies a recursive definition of the reduced basis

$$\Phi_{k_{m+1}}^{(m)}(x, y_1, \dots, y_m) = \sum_{k_m=1}^{r_m} \sum_{\mu_m=0}^{d_m-1} U_m(k_m, \mu_m, k_{m+1}) \Phi_{k_m}^{(m-1)}(x, y_1, \dots, y_{m-1}) P_{\mu_m}(y_m) \tag{3.7}$$

and

$$U_{\leq m}(i, \mu_1, \dots, \mu_m, k_{m+1}) = \sum_{k_m=1}^{r_m} U_{\leq m-1}(i, \mu_1, \dots, \mu_{m-1}, k_m) U_m(k_m, \mu_m, k_{m+1}). \tag{3.8}$$

In practice, it is convenient to orthogonalize the basis functions in (3.4). In the present case of parametric boundary value problems, we require the orthogonality of the first set of reduced basis functions $\{\Phi_{k_1}^{(0)} : k_1 = 1, \dots, r_1\}$ with respect to an inner product equivalent to the $H^1(D)$-inner product. The basis sets for all other $\mathcal{L}_m$ are orthogonalized w.r.t. an inner product for a Hilbert space norm equivalent to the $H^1(D) \otimes L^2_{\pi_1}([-1, 1])^M$ norm, see (2.14).

In the same fashion, we reduce the basis of the remaining space in $\mathcal{L}_m \otimes \mathcal{Y}_{m+1} \otimes \ldots \otimes \mathcal{Y}_M$, obtaining subspaces

$$\mathcal{R}_{m+1} := \operatorname{span}\{\Psi_{k_{m+1}}^{(m+1)} : k_{m+1} = 1, \dots, r_{m+1}\}, \tag{3.9}$$

$$\mathcal{R}_{m+1} \subset \mathcal{Y}_{m+1} \otimes \ldots \otimes \mathcal{Y}_M \tag{3.10}$$

with basis functions

$$\begin{aligned}&\Psi_{k_{m+1}}^{(m+1)}(y_{m+1},\dots,y_M)\\ &\quad=\sum_{\mu_{m+1}=0}^{d_{m+1}-1}\cdots\sum_{\mu_M=0}^{d_M-1}U_{\geq m+1}(k_{m+1},\mu_{m+1},\dots,\mu_M)P_{\mu_{m+1}}(y_{m+1})\cdots P_{\mu_M}(y_M).\end{aligned}\tag{3.11}$$

The original problem then reduces to a much smaller problem with solution

$$u_N(x,y)=\sum_{k_{m+1}=1}^{r_{m+1}}\Phi_{k_{m+1}}^{(m)}(x,y_1,\dots,y_m)\Psi_{k_{m+1}}^{(m+1)}(y_{m+1},\dots,y_M)\tag{3.12}$$

depending only on the (discrete) parameter k_{m+1}. We often require the reduced basis functions in $\mathcal{R}_{m+1}$ to be orthogonal as well. This motivates the ALS algorithm defined below.

We mention in passing that the reduced basis of the first component of the tensor (3.1) is constructed implicitly when evaluating the solution tensor with the ALS of Section 4. Since this basis can immediately be extracted and used for further computations, the suggested approach could thus be understood as the offline phase of a reduced basis method.

3.2. **The Tensor Train Format.** Expanding on the recursive formula (3.8), we obtain an explicit decomposition of any tensor $U\in\mathbb{R}^{N\times d_1\times\cdots\times d_M}$ of the form

$$U(i,\mu_1,\dots,\mu_M)=\sum_{k_1=1}^{r_1}\cdots\sum_{k_{M+1}=1}^{r_{M+1}}U_0(i,k_1)\prod_{m=1}^{M}U_m(k_m,\mu_m,k_{m+1}),\tag{3.13}$$

where $r_{m+1}=1$. Since the r_m are exactly the dimensions of the subspaces $r_m=\dim\mathcal{L}_m$, this is just the TT representation that was introduced by Oseledets et al. in [54]. It has received much attention in recent years and it has already been applied to stochastic PDEs in [16].

Throughout this paper, the discrete solution u_N will be represented by a TT tensor U as in (2.34). While this is a further approximation, we circumvent the curse of dimensionality in exchange since this tensor grows only linearly in M. The question remains whether this approach is justified and accurate enough in the present case. In general, a good approximation can be achieved if the decay of the singular values of every unfolding of U is fast, see e.g. [44]. This is often given if the solution u exhibits very high regularity which cannot be guaranteed here. However, in our context, the quality of the approximation only needs to be on par with the other approximations. Improvement is necessary only for very fine meshes and a large tensor set Λ. Therefore, it is very reasonable to reduce the basis sets accordingly.

We define the left and right unfoldings

$$U_m^{\mathsf{L}}=[U_m]_{k_m\mu_m}^{k_{m+1}}\in\mathbb{R}^{r_md_m\times r_{m+1}}\quad\text{and}\quad U_m^{\mathsf{R}}=[U_m]_{k_m}^{\mu_mk_{m+1}}\in\mathbb{R}^{r_m\times d_mr_{m+1}}\tag{3.14}$$

as the two matricization of the components, respectively. The above orthogonality is accomplished by requiring

$$(U_m^{\mathsf{L}})^{\mathsf{T}}U_m^{\mathsf{L}}=I_{r_{m+1}}.\tag{3.15}$$

The component U_m is then called *left-orthogonal*. For U_0 we also require

$$U_0^{\mathsf{T}}U_0=I_{r_1}\tag{3.16}$$

for now, but this will be modified later. U_m is called *right-orthogonal* if

$$U_m^{\mathsf{R}}(U_m^{\mathsf{R}})^{\mathsf{T}} = I_{r_m}. \tag{3.17}$$

Orthogonalization can be achieved by performing an SVD on the component U_m and shifting parts to the next component U_{m+1}, i.e., for left-orthogonality,

$$U_m^{\mathsf{L}} = X_m \Sigma_m Y_m^{\mathsf{T}}, \tag{3.18}$$

$$U_m^{\mathsf{L}} \leftarrow X_m, \tag{3.19}$$

$$U_{m+1}^{\mathsf{R}} \leftarrow \Sigma_m Y_m^{\mathsf{T}} U_{m+1}^{\mathsf{R}}. \tag{3.20}$$

3.3. **Calculating Mean and Variance in TT Format.** The TT format allows to compute and postprocess the whole parametric solution of problem (2.6) efficiently. For any given $x \in D$, $y \in [-1,1]^M$, this would conventionally be calculated by evaluating the full sum (2.34). Computational costs for obtaining this representation grows exponentially with dimension M but this can be overcome by using the TT structure of the tensor U as in (3.13). Notice that u_N can be rewritten as

(3.21)
$$u_N(x,y) = \sum_{k_1=1}^{r_1} \cdots \sum_{k_M=1}^{r_M} \left(\sum_{i=0}^{N-1} U_0(i,k_1)\varphi_i(x) \right) \prod_{m=1}^{M} \left(\sum_{\mu_m=0}^{d_m-1} U_m(k_m,\mu_m,k_{m+1}) P_{\mu_m}(y_m) \right).$$

It can be seen directly that computational cost now is in fact linear in M. This extends to the calculation of stochastic moments, e.g. the mean and the variance. For the mean, we get

$$\mathbb{E}(u_N) = \int_\Gamma u_N(y)\,\mathrm{d}\pi(y) \tag{3.22}$$

$$= \sum_{i=0}^{N-1} \sum_{\mu\in\Lambda} U(i,\mu)\varphi_i \prod_{m=1}^{M} \int_{-1}^{1} P_{\mu_m}(y_m)\,\mathrm{d}\pi(y_m) \tag{3.23}$$

$$= \sum_{i=0}^{N-1} U(i,0,\ldots,0)\varphi_i \tag{3.24}$$

due to the orthonormality of $(P_{\mu_m})_{\mu_m=0}^{\infty}$ in $L^2_{\pi_1}([-1,1])$. The variance defined by

$$\mathbb{V}(u_N) = \mathbb{E}((u_N - \mathbb{E}(u_N))^2) \tag{3.25}$$

$$= \mathbb{E}(u_N^2) - (\mathbb{E}(u_N))^2 \tag{3.26}$$

can be evaluated similarly. Here, the second part is just the square of the mean as above and the first part is given by

$$\mathbb{E}(u_N^2) = \int_\Gamma u_N^2(y)\,\mathrm{d}\pi(y) \tag{3.27}$$

$$= \int_\Gamma \sum_{i,j=0}^{N-1} \sum_{\mu,\nu\in\Lambda} U(i,\mu)U(j,\nu)\varphi_i\varphi_j P_\mu(y)P_\nu(y)\,\mathrm{d}\pi(y) \tag{3.28}$$

$$= \sum_{i,j=0}^{N-1} \sum_{\mu\in\Lambda} U(i,\mu)U(j,\mu)\varphi_i\varphi_j \tag{3.29}$$

because of the orthogonality of the P_μ. If the solution is right-orthogonal (3.17), this can be simplified to

$$\mathbb{E}(u^2) = \sum_{i,j=0}^{N-1} \sum_{k_1=1}^{r_1} U_0(i,k_1)U_0(j,k_1)\varphi_i\varphi_j. \tag{3.30}$$

Note that also higher moments can be calculated based on the preceding derivations.

3.4. **Fixed-Rank Manifolds and Varieties.** The representation (3.13) is called a *TT decomposition* if $r = (r_1, \dots, r_M)$ is minimal in each component. It can be shown that such a minimal decomposition always exists and is computable in polynomial time [41]. r is then called the *TT-rank* of the decomposition.

We consider the spaces

$$\mathcal{M}_r = \{U \in \mathbb{R}^{N\times d_1\times\dots\times d_M} : U \text{ has TT-rank } r\}, \tag{3.31}$$

$$\mathcal{M}_{\leq r} = \{U \in \mathbb{R}^{N\times d_1\times\dots\times d_M} : U \text{ has TT-rank smaller or equal to } r\}. \tag{3.32}$$

It has been established in [41] and [64] that $\mathcal{M}_r$ is a smooth manifold which is embedded in the tensor space. In [30] it has been shown that $\mathcal{M}_{\leq r}$ is an algebraic variety, i.e. the set of common zeros of some multi-variate polynomials, and as such it is closed. This is important for optimization tasks and does not hold for all tensor decomposition formats such as e.g. the canonical format [14]. A crucial drawback is that the manifold $\mathcal{M}_r$ is in general not connected and highly non-convex. This will be discussed in the next section.

If we fix r and consider only tensors $U \in \mathcal{M}_{\leq r}$, we can define the space of components $\mathcal{C}_{\leq r} = \{(U_0, U_1, \dots, U_M) : U_0 \in \mathbb{R}^{N\times r_1},\ U_m \in \mathbb{R}^{r_{m-1}\times d_m\times r_m}\}$ and the map

$$\tau : \mathcal{C}_{\leq r} \ \to \ \mathcal{M}_{\leq r}, \tag{3.33}$$

$$(U_0, U_1, \dots, U_M) \ \mapsto \ \tau(U_0, U_1 \dots, U_M) := U, \tag{3.34}$$

for the introduction of the ALS algorithm in the next section. Note that this map is not injective. It can be made so by imposing a quotient structure on $\mathcal{C}_{\leq r}$ as discussed in [64] but this is not of interest here.

4. The ALS Algorithm

This section is devoted to the description of the employed solution algorithm. While there are many advanced algorithms available, we employ a very simple iteration procedure to obtain a low-rank approximation of the solution. The description provides some understanding of the nature of the present tensor approximation and its relation to reduced basis functions.

Upon recalling the definitions of Section 2.4, in the given problem setting, we minimize the functional

$$J(U) := \frac{1}{2}\langle \mathbf{A}(U), U\rangle - \langle F, U\rangle. \tag{4.1}$$

This can be defined equivalently on the component space $\mathcal{C}_{\leq r}$ as

$$j(U_0, U_1, \dots, U_M) := J(\tau(U_0, U_1, \dots, U_M)). \tag{4.2}$$

To minimize this functional, we introduce the Alternating Least Squares (ALS) algorithm: For $m \in \{0, \dots, d\}$, fix $U_0, \dots, U_{m-1}$ and $U_{m+1}, \dots, U_M$ and solve the subproblem

$$\min_{V_m \in \mathbb{R}^{r_m \times d_m \times r_{m+1}}} j(U_0, \dots, V_m, \dots, U_M) \tag{4.3}$$

in a least squares sense. This can be done in a successive manner and with alternating directions which justifies the name of the procedure.

The TT format allows for an explicit formulation of this algorithm, sometimes dubbed the *Alternating Linear Scheme*, to maintain the abbreviation. In this case, we can derive a closed form for each subproblem. These can be solved using standard tools from linear algebra and numerical optimization.

For fixed $m = 1, \dots, M-1$, we solve the subproblem on the subspace

$$\mathcal{L}_{m-1} \otimes \mathcal{Y}_m \otimes \mathcal{R}_{m+1} \simeq \mathbb{R}^{r_m \times d_m \times r_{m+1}} \tag{4.4}$$

via

$$\begin{aligned} u_N(x,y) = &\sum_{k_m=1}^{r_m} \sum_{\mu_m=0}^{d_m-1} \sum_{k_{m+1}=1}^{r_{m+1}} U_m(k_m, \mu_m, k_{m+1}) \\ &\times \Phi_{k_m}^{(m-1)}(x, y_1, \dots, y_{m-1}) P_{\mu_m}(y_m) \Psi_{k_{m+1}}^{(m+1)}(y_{m+1}, \dots, y_M). \end{aligned} \tag{4.5}$$

For $m = 0$, this subspace becomes $\mathcal{X}_p(\mathcal{T}) \otimes \mathcal{R}_1 \simeq \mathbb{R}^{N \times r_1}$ and for $m = M$ we get $\mathcal{L}_{M-1} \otimes \mathcal{Y}_M \simeq \mathbb{R}^{r_M \times d_M}$ and (4.5) changes accordingly.

We solve the subproblem by projecting the tensor $U \in \mathbb{R}^{N \times d_1 \times \dots \times d_M}$ onto these subspaces using the orthogonal projection $\mathbf{P}_m$. If we choose the reduced basis functions $\{\Phi_{k_m}^{(m-1)} : k_m = 1, \dots, r_m\}$ and $\{\Psi_{k_{m+1}}^{(m+1)} : k_{m+1} = 1, \dots, r_{m+1}\}$ to be orthogonal, this projection is given by the insertion operators $\mathbf{E}_m(U_m) = U$ and $\mathbf{E}_m^\dagger(U) = U_m$ as introduced in [42], characterized by

$$\mathbf{P}_m = \mathbf{E}_m \mathbf{E}_m^\dagger. \tag{4.6}$$

Because of the orthogonality, we also get $\mathbf{E}_m^\dagger \mathbf{E}_m = \mathbf{id}$. We remark that a sufficient condition for the orthogonality of $\{\Phi_{k_m}^{(m-1)} : k_m = 1, \dots, r_m\}$ and $\{\Psi_{k_{m+1}}^{(m+1)} : k_{m+1} = 1, \dots, r_{m+1}\}$ is maintained if $U_0, \dots, U_{m-1}$ are left-orthogonal and $U_{m+1}, \dots, U_M$ are right-orthogonal. Thus, it suffices to shift orthogonality to the next component after the optimization step in U_m in order to prepare for the next step.

The subproblem in (4.2) becomes

$$j(U_0, U_1, \dots, U_M) = \frac{1}{2} \langle \mathbf{A}\mathbf{E}_m(U_m), \mathbf{E}_m(U_m) \rangle - \langle F, \mathbf{E}_m(U_m) \rangle \tag{4.7}$$

$$= \frac{1}{2} \langle \mathbf{E}_m^\dagger \mathbf{A} \mathbf{E}_m(U_m), U_m \rangle - \langle \mathbf{E}_m^\dagger(F), U_m \rangle, \tag{4.8}$$

which yields

$$U_m^+ = \operatorname{argmin}_{V_m \in \mathbb{R}^{r_m \times d_m \times r_{m+1}}} j(U_0, \dots, V_m, \dots, U_M) \tag{4.9}$$

$$= (\mathbf{E}_m^\dagger \mathbf{A} \mathbf{E}_m)^{-1} \mathbf{E}_m^\dagger(F), \tag{4.10}$$

$$U^+ = (\mathbf{P}_m \mathbf{A} \mathbf{P}_m)^{-1}(F) \tag{4.11}$$

for every iteration step m.

It is important to note that the rank has to be chosen in advance since the functional j is defined on a fixed component space $\mathcal{C}_{\leq r}$. Hence, one usually would reduce the ranks after some iteration steps. This means that we minimize on the manifold $\mathcal{M}_r$ or the variety $\mathcal{M}_{\leq r}$. Several approaches to rank adaptivity have been introduced already. The most well-known might be the DMRG algorithm that is mostly used in quantum physics [62]. We present a more recent method in section 6.

Convergence results for the ALS algorithm are subject to current research and so far only local convergence can be assumed [58, 25, 60]. If the ALS converges to a point U^*, it was shown that this is a stationary point [42]. Therefore, if J is a convex functional and we choose full rank r, there is convergence to a global minimum. However, since the manifolds $\mathcal{M}_r$ are non-convex, this does not hold for smaller ranks. In this case, we can hope to find a local minimum at best.

Nevertheless, in practice, the ALS algorithm often outperforms more sophisticated algorithms significantly and remains the "workhorse" algorithm in tensor optimization [46].

4.1. **Preconditioning.** Preconditioning is key to the convergence behavior of the solver. We choose to precondition the functional j using the mean value of the coefficient such that $\mathbf{A}_0 = K_0 \otimes I \otimes \cdots \otimes I$ as for instance discussed in [31, 57, 56, 63]. Thus, for each m, we are minimizing

$$j(U_0, \dots, V_m, \dots, U_M) = \frac{1}{2}\langle \mathbf{C}\mathbf{A}\mathbf{C}^\dagger \mathbf{E}_m(V_m), \mathbf{E}_m(V_m)\rangle - \langle \mathbf{C}^\dagger(F), \mathbf{E}_m(V_m)\rangle, \tag{4.12}$$

where $\mathbf{A}_0^{-1} = (K_0)^{-1} \otimes I \otimes \cdots \otimes I =: CC^\mathsf{T} \otimes I \otimes \cdots \otimes I =: \mathbf{C}\mathbf{C}^\dagger$ is the unique Cholesky decomposition. For $m = 1, \dots, M$, this can be incorporated into the insertion operators,

$$\tilde{U}_0 = C^\mathsf{T} U_0, \tag{4.13}$$

$$\tilde{\mathbf{E}}_m = \mathbf{C}^\dagger \mathbf{E}_m. \tag{4.14}$$

However, this would require an explicit computation of the inverse K_0^{-1} as well as its Cholesky decomposition. Therefore, we exploit that for left-orthogonal U_0, (4.13) is equivalent to

$$(\tilde{U}_0^\mathsf{L})^\mathsf{T} K_0 \tilde{U}_0^\mathsf{L} = I_{r_1}, \tag{4.15}$$

i.e., we generally require the first component to be orthogonal with respect to the K_0-inner product. This yields exactly the inner product (2.14), i.e. we obtain

$$(a_0 \Phi_{k_1}, \Phi_{k_1'})_\mathcal{X} = \delta_{k_1 k_1'}. \tag{4.16}$$

Subsequently, we can rewrite (4.12) as

$$j(U_0, \dots, V_m, \dots, U_M) = \frac{1}{2}\langle \mathbf{A}\tilde{\mathbf{E}}_m(V_m), \tilde{\mathbf{E}}_m(V_m)\rangle - \langle F, \tilde{\mathbf{E}}_m(V_m)\rangle. \tag{4.17}$$

This means, by ensuring orthogonality of the first component w.r.t. the mean inner product, we can incorporate the preconditioner effectively into the ALS-step. An explicit preconditioning has only to be carried out for the first subproblem.

4.2. **Rank adaptivity.** As explained above, the ALS algorithm finds a stationary point in a subset $\mathcal{M}_{\leq r}$ of the fully discretized tensor space $\mathbb{R}^{N \times d_1 \times \dots \times d_M}$. Since this set is an algebraic variety, a

necessary and sufficient condition for such a stationary point is that the projection of the gradient onto the tangent cone

(4.18)
$$T_U\mathcal{M}_{\leq r} = \{\xi \in \mathbb{R}^{N\times d_1\times\cdots\times d_M} : \exists (U^n) \subseteq \mathcal{M}_{\leq r}, (\alpha^n) \subseteq \mathbb{R}^+ \text{ s.t. } U^n \to U, \alpha^n(U^n - U) \to \xi\}$$

vanishes, see e.g. [60]. In other words, U^* is a stationary point of the functional J if

(4.19)
$$-\nabla J(U^*) \in T^\circ_{U^*}\mathcal{M}_{\leq r},$$

where

(4.20)
$$T^\circ_U\mathcal{M}_{\leq r} = \{y \in \mathbb{R}^{N\times d_1\times\ldots\times d_M} : \langle y, \xi\rangle \leq 0 \text{ for all } \xi \in T_U\mathcal{M}_{\leq r}\}$$

is the polar tangent cone of regular normal vectors at U. In the case where we optimize only on the embedded manifold $\mathcal{M}_r$, the tangent cone equals the linear tangent space of the manifold and the polar tangent cone is its orthogonal complement, i.e. the normal space.

Since we know that the ALS algorithm results in a stationary point $U^* \in \mathcal{M}_{\leq r}$ if it converges, we can deduce that the discrete residual $\mathbf{R}(U^*) := \mathbf{A}(U^*) - F = \nabla J(U^*)$ lies completely in the polar tangent cone at U^*. Thus, any step taken in gradient direction will lead us off the variety $\mathcal{M}_{\leq r}$ and hence increase the rank.

This motivates a straightforward strategy for rank increase as discussed in [19]: Find an appropriate rank-k approximation of the discrete residual and add it to the tensor U^*. For example, a rank-1 update will increase every rank by at most 1 and thus yield a sizable enlargement of the search space $\mathcal{M}_{\leq r}$. According to [34] such a rank-1 approximation is given by truncating the singular values of every unfolding.

Similar to [20, 21], we introduce an estimator for the error in the full tensor space in Section 6. If the ALS error outweighs the other error components which are the result of various discretizations, a rank increase of the tensor is performed. Note that we aim at keeping the rank comparatively low since increasing it results in an immediate and significant enlargement of the computational complexity of the problem.

5. Error Estimates

In this section, we recall an error estimator for the overall mean square energy error which was derived in [20, 21]. It consists of different error contributions which value the discretization and stochastic truncation errors as well as a consistency error, in our case arising from inexact solves in low-rank format. In addition to a bound for the consistency term, our fundamental contribution is the highly efficient evaluation of the error terms based on the exploitation of the TT format. With this, as with the solution of the discrete problem, the curse of dimensionally can be avoided.

Following [20], we define the stochastic tail indicator for $w_\Lambda \in \mathcal{V}(\Lambda)$,

(5.1)
$$\zeta_\nu(w_\Lambda) := \sum_{m=1}^{\infty} \left\| \frac{a_m}{a_0} \right\|_{L^\infty(D)} (\beta_{\nu_m+1} \|w_{\Lambda,\nu+\epsilon_m}\|_{\mathcal{X}} + \beta_{\nu_m} \|w_{\Lambda,\nu-\epsilon_m}\|_{\mathcal{X}}),$$

and the deterministic residual-based error estimator for any $w_N \in \mathcal{V}_p(\Lambda;\mathcal{T})$,

(5.2)
$$\eta_\mu(w_N)^2 := \sum_{T\in\mathcal{T}} \eta_{\mu,T}(w_N)^2 + \sum_{S\in\mathcal{S}} \eta_{\mu,S}(w_N)^2$$

with volume and edge contributions on $T \in \mathcal{T}$ and $S \in \mathcal{S}$ given by

$$\eta_{\mu,T}(w_N)^2 := h_T^2 \|a_0^{-1/2}(f\delta_{\mu 0} + \nabla \cdot \sigma_\mu(w_N))\|_{L^2(T)}^2, \tag{5.3}$$

$$\eta_{\mu,S}(w_N)^2 := h_S \|a_0^{-1/2}[\![\sigma_\mu(w_N)]\!]_S\|_{L^2(S)}^2. \tag{5.4}$$

Here, the numerical flux $\sigma_\mu(w_N)$ is defined by

$$\sigma_\mu(w_N) := a_0 \nabla w_{N,\mu} + \sum_{m=1}^{\infty} a_m \nabla(\beta_{\mu_m+1} w_{N,\mu+\epsilon_m} + \beta_{\mu_m} w_{N,\mu-\epsilon_m}). \tag{5.5}$$

The stochastic and deterministic upper bounds are then given by $\zeta(w_\Lambda)^2 := \sum_{\nu\in\mathcal{F}} \zeta_\nu(w_\Lambda)^2$ and $\eta(w_N)^2 := \sum_{\mu\in\Lambda} \eta_\mu(w_N)^2$, respectively. With this in place, the following overall error bound can be derived.

Theorem 5.1 ([20] Thm. 6.2)**.** *For any $w_N \in \mathcal{V}_p(\Lambda;\mathcal{T})$, the solution $u \in \mathcal{V}$ of* (2.1) *and the Galerkin approximation $u_N \in \mathcal{V}_p(\Lambda;\mathcal{T})$ in* (2.32)*, it holds*

$$\begin{aligned}\|w_N - u\|_{\mathcal{A}}^2 \le \eta_{\mathcal{A}} := \Bigg(&\frac{c_\eta}{\sqrt{1-\gamma}}\eta(w_N) + \frac{c_Q}{\sqrt{1-\gamma}}\zeta(w_N) \\ &+ c_Q\|w_N - u_N\|_{\mathcal{A}}\Bigg)^2 + \|w_N - u_N\|_{\mathcal{A}}^2.\end{aligned} \tag{5.6}$$

Remark 5.2. For our purposes, c_Q denotes the operator norm of $\mathrm{id} - \mathcal{I}$ with $\mathcal{I}$ the Clément interpolation operator, and c_η stems from interpolation constants and a uniform overlap condition due to the regular grid $\mathcal{T}$. More details on the definition of these constants can be found in [20].

The first sum in (5.6) suffers from the curse of dimensionality as we need to sum over the error estimates for each $\mu \in \Lambda$. The second sum in (5.6) is infinite but each term can be computed very quickly. We assume the sum can in principle be evaluated (up to very high accuracy). We address these issues independently in the following sections.

5.1. **Tail Estimator.** To efficiently evaluate the stochastic tail estimator (5.1), it is beneficial to sum over all indices which affect a certain stochastic dimension n, since we aim at maintaining the full tensor set. We thereby obtain an estimator for any component $n = 1,\dots,M$ (and analogously for all inactive components $n > M$).

Let

$$\begin{aligned}\Delta_n &:= \{\nu \in \partial\Lambda : \nu - \epsilon_n \in \Lambda\} \\ &= \{(\nu_1,\dots,d_n-1,\dots,\nu_M,0,\dots) \in \mathcal{F} \mid \\ &\qquad \nu_m = 0,\dots,d_m-1;\ m = 1,\dots,\cancel{n},\dots,M\}.\end{aligned} \tag{5.7}$$

(5.8)

Then, the estimator for the n-th component subject to some (semi-discrete) $w_\Lambda \in \mathcal{V}(\Lambda)$ is given by

$$\zeta_n(w_\Lambda)^2 = \sum_{\nu \in \Delta_n} \zeta_\nu(w_\Lambda)^2 \tag{5.9}$$

$$= \sum_{\nu \in \Delta_n} \left(\sum_{m=1}^{\infty} \left\| \frac{a_m}{a_0} \right\|_{L^\infty(D)} \left(\beta_{\nu_m+1} \left\| w_{\Lambda,\nu+\epsilon_m} \right\|_{\mathcal{X}} + \beta_{\nu_m} \left\| w_{\Lambda,\nu-\epsilon_m} \right\|_{\mathcal{X}} \right) \right)^2 \tag{5.10}$$

$$= \sum_{\nu \in \Delta_n} \left(\left\| \frac{a_n}{a_0} \right\|_{L^\infty(D)} \beta_{\nu_n} \left\| w_{\Lambda,\nu-\epsilon_n} \right\|_{\mathcal{X}} \right)^2 \tag{5.11}$$

$$= \left(\beta_{d_n-1} \left\| \frac{a_n}{a_0} \right\|_{L^\infty(D)} \right)^2 \sum_{\nu \in \Delta_n} \left\| w_{\Lambda,\nu-\epsilon_n} \right\|_{\mathcal{X}}^2 \ . \tag{5.12}$$

Here, we use the fact that $w_{\Lambda,\nu+\epsilon_m} \equiv 0$ for all m since $\nu \in \partial\Lambda$. For the same reason we get $w_{\Lambda,\nu-\epsilon_m} \equiv 0$ for all $m \neq n$. By decomposing $w_{\Lambda,\nu} = \sum_{i=0}^{N-1} W(i,\nu)\varphi_i$ with the discrete FE basis $\{\varphi_j\}_{j=0}^{N-1}$, the last sum can be written as

$$\sum_{\nu \in \Delta_n} \left\| w_{\Lambda,\nu-\epsilon_n} \right\|_{\mathcal{X}}^2 = \sum_{\nu \in \Delta_n} (w_{\Lambda,\nu-\epsilon_n}, w_{\Lambda,\nu-\epsilon_n})_{\mathcal{X}} \tag{5.13}$$

$$= \sum_{\nu \in \Delta_n} \sum_{i,j=0}^{N-1} (W(i,\nu-\epsilon_n)\varphi_i, W(j,\nu-\epsilon_n)\varphi_j)_{\mathcal{X}} \tag{5.14}$$

$$= \sum_{i,j=0}^{N-1} \sum_{\nu \in \Delta_n} W(i,\nu-\epsilon_n) W(j,\nu-\epsilon_n)(\varphi_i,\varphi_j)_{\mathcal{X}}. \tag{5.15}$$

The sum over Δ_n can be seen as the trace over all stochastic indices with the n-th index fixed from which follows

$$\sum_{\nu \in \Delta_n} W(i,\nu-\epsilon_n) W(j,\nu-\epsilon_n) = \tag{5.16}$$
$$\sum_{\mu_1=0}^{d_1-1} \cdots \cancel{\sum_{\mu_n=0}^{d_n-1}} \cdots \sum_{\mu_M=0}^{d_M-1} W(i,\mu_1,\ldots,d_n-1,\ldots,\mu_M) W(j,\mu_1,\ldots,d_n-1,\ldots,\mu_M).$$

This can be calculated efficiently in the TT format and does not scale exponentially. We obtain a low-rank decomposition as in (3.30)

$$\sum_{\nu \in \Delta_n} W(i,\nu-\epsilon_n) W(j,\nu-\epsilon_n) = \sum_{k_1=1}^{r_1} \sum_{k_1'=1}^{r_1} W_0(i,k_1) R(k_1,k_1') W_0(j,k_1'), \tag{5.17}$$

where

$$R(k_1,k_1') = \tag{5.18}$$
$$\sum_{\mu_1=0}^{d_1-1} \cdots \cancel{\sum_{\mu_n=0}^{d_n-1}} \cdots \sum_{\mu_M=0}^{d_M-1} W_{\geq 1}(k_1,\mu_1,\ldots,d_n-1,\ldots,\mu_M) W_{\geq 1}(k_1',\mu_1,\ldots,d_n-1,\ldots,\mu_M).$$

Storage and computational cost is not independent of the size of Λ and depends only on the much smaller rank r_1. In accordance with [20], for any $n \in \mathbb{N} \setminus \operatorname{supp}(\Lambda)$, (5.12) becomes

$$\zeta_n(w_\Lambda)^2 = \left(\beta_1 \left\|\frac{a_n}{a_0}\right\|_{L^\infty(D)}\right)^2 \sum_{\nu\in\Lambda} \|w_{\Lambda,\nu}\|_{\mathcal{X}}^2\,, \tag{5.19}$$

with β_1 constant and $\omega := \sum_{\nu\in\Lambda} \|w_{\Lambda,\nu}\|_{\mathcal{X}}^2$ which can be calculated efficiently as above. The second sum in (5.6) then becomes

$$\sum_{\mu\in\mathcal{F}} \zeta_\mu(w_N)^2 = \sum_{n\in\operatorname{supp}(\Lambda)} \zeta_n(w_N)^2 + \sum_{n\in\mathbb{N}\setminus\operatorname{supp}(\Lambda)} \zeta_n(w_N)^2 \tag{5.20}$$

$$= \sum_{n\in\operatorname{supp}(\Lambda)} \zeta_n(w_N)^2 + \beta_1^2\omega \sum_{n\in\mathbb{N}\setminus\operatorname{supp}(\Lambda)} \left\|\frac{a_n}{a_0}\right\|_{L^\infty(D)}^2. \tag{5.21}$$

Assumption (2.11) ensures convergence of

$$\sum_{n\in\mathbb{N}\setminus\operatorname{supp}(\Lambda)} \left\|\frac{a_n}{a_0}\right\|_{L^\infty(D)} \tag{5.22}$$

and thus the sum in the error estimate is finite.

5.2. **Deterministic Estimator.** We can exploit the TT structure for the deterministic error estimator as in the preceding section. Assume some (fully discrete) $w_N \in \mathcal{V}_p(\Lambda;\mathcal{T})$. Since all sums in the expression (5.2) are finite, for the first part we deduce

$$\sum_{\mu\in\Lambda} \eta_{\mu,T}(w_N)^2 = \sum_{\mu\in\Lambda} h_T^2 \langle f\delta_{\mu 0} + \nabla\cdot\sigma_\mu(w_N), a_0^{-1}(f\delta_{\mu 0} + \nabla\cdot\sigma_\mu(w_N))\rangle_T \tag{5.23}$$

$$= h_T^2\big(\langle f\delta_{\mu 0}, a_0^{-1} f\rangle_T + 2\langle f\delta_{\mu 0}, a_0^{-1}\nabla\cdot\sigma_0(w_N)\rangle_T$$

$$+ \sum_{\mu\in\Lambda} \langle \nabla\cdot\sigma_\mu(w_N), a_0^{-1}\nabla\cdot\sigma_\mu(w_N)\rangle_T\big). \tag{5.24}$$

As before, taking the sum over all $\mu \in \Lambda$ is computationally too complex. Instead, we exploit the structure of the approximation $w_N = \sum_{i=0}^{N-1}\sum_{\mu\in\Lambda} W(i,\mu)\varphi_i P_\mu$ which yields

$$\sigma_\mu(w_N) = \sum_{m=0}^{M}\sum_{i=0}^{N-1} a_m \nabla\varphi_i W^m(i,\mu), \tag{5.25}$$

where we set

$$W^m := \big[I_0 \otimes I_1 \otimes \cdots \otimes B_m \otimes \cdots \otimes I_M\big](W) \tag{5.26}$$

and B_m is as in (2.36) with the finite index set Λ instead of $\mathcal{F}$. We obtain for the last term of (5.23), on any $T \in \mathcal{T}$,

$$\sum_{\mu\in\Lambda} \langle \nabla \cdot \sigma_\mu(w_N), a_0^{-1} \nabla \cdot \sigma_\mu(w_N)\rangle_T$$

$$= \sum_{\mu\in\Lambda} \langle \nabla \cdot \sum_{m_1=0}^{M} \sum_{i=0}^{N-1} a_{m_1} \nabla\varphi_i W^{m_1}(i,\mu), a_0^{-1}\nabla \cdot \sum_{m_2=0}^{M} \sum_{j=0}^{N-1} a_{m_2}\nabla\varphi_j W^{m_2}(j,\mu)\rangle_T \tag{5.27}$$

$$= \sum_{m_1,m_2=0}^{M} \sum_{i,j=0}^{N-1} \langle \nabla\cdot(a_{m_1}\nabla\varphi_i), a_0^{-1}\nabla\cdot(a_{m_2}\nabla\varphi_j)\rangle_T \sum_{\mu\in\Lambda} W^{m_1}(i,\mu)W^{m_2}(j,\mu). \tag{5.28}$$

Since W^m is a tensor in TT format, the sum over all $\mu \in \Lambda$ can again be evaluated in polynomial (see (3.21)) and the curse of dimensionality does not incur. Additionally, similar to (5.17), it can be advantageous to perform a low-rank decomposition of W^m such that

$$\sum_{\mu\in\Lambda} W^{m_1}(i,\mu)W^{m_2}(j,\mu) = \sum_{k_1=1}^{r_1}\sum_{k_1'=1}^{r_1} W_0(i,k_1)R^{m_1,m_2}(k_1,k_1')W_0(j,k_1'). \tag{5.29}$$

Observe that the dependence on m_1, m_2 is encoded entirely in the middle term R^{m_1,m_2}. It can be incorporated into either the left or the right term, e.g.

$$V^{m_1,m_2}(i,k_1') := \sum_{k_1=1}^{r_1} W_0(i,k_1)R^{m_1,m_2}(k_1,k_1'). \tag{5.30}$$

We thus obtain

$$\sum_{\mu\in\Lambda} \langle \nabla \cdot \sigma_\mu(w_N), a_0^{-1} \nabla \cdot \sigma_\mu(w_N)\rangle_T =$$

$$\sum_{m_1,m_2=0}^{M} \sum_{k_1=1}^{r_1} \langle \nabla \cdot \sum_{i=0}^{N-1} a_{m_1}\nabla\varphi_i V^{m_1,m_2}(i,k_1), a_0^{-1}\nabla\cdot\sum_{j=0}^{N-1} a_{m_2}\nabla\varphi_j W_0(j,k_1)\rangle_T. \tag{5.31}$$

This retains the original format that can be solved easily in the finite element setting. However, the sum over all $\mu \in \Lambda$ where $|\Lambda|$ grows exponentially with M is replaced by a sum over the first rank $k_1 = 1, \dots, r_1$, and two sums over M which are much more efficient to evaluate.

The second part of the estimator can be calculated analogously on any edge $S \in \mathcal{S}$ by

$$\sum_{\mu\in\Lambda} \eta_{\mu,S}(w_N)^2 = \sum_{\mu\in\Lambda} h_S \| a_0^{-1/2} [\![\sigma_\mu(w_N)]\!]_S \|^2_{L^2(T)} \tag{5.32}$$

$$= \sum_{\mu\in\Lambda} h_S \langle [\![\sigma_\mu(w_N)]\!]_S, a_0^{-1}[\![\sigma_\mu(w_N)]\!]_S\rangle_S \tag{5.33}$$

$$= h_S \sum_{m_1,m_2=0}^{M}\sum_{i,j=0}^{N-1} \langle [\![a_{m_1}\nabla\varphi_i]\!]_S, a_0^{-1}[\![a_{m_2}\nabla\varphi_j]\!]_S\rangle_S$$

$$\times \sum_{\mu\in\Lambda} W^{m_1}(i,\mu)W^{m_2}(j,\mu). \tag{5.34}$$

It follows that

$$\sum_{\mu\in\Lambda}\eta_{\mu,S}(w_N)^2 =$$

$$h_S \sum_{m_1,m_2=0}^{M}\sum_{k_1=1}^{r_1}\langle[\![\sum_{i=0}^{N-1} a_{m_1}\nabla\varphi_i V^{m_1,m_2}(i,k_1)]\!]_S, a_0^{-1}[\![\sum_{j=0}^{N-1} a_{m_1}\nabla\varphi_j W_0(j,k_1)]\!]_S\rangle_S. \tag{5.35}$$

5.3. **ALS Residual.** For the last component of (5.6), we estimate the distance to the best approximation in the energy norm by using the discrete residual $r_N(w_N) := \mathcal{A}(w_N - u_N)$,

$$\|w_N - u_N\|_{\mathcal{A}}^2 = \int_\Gamma \langle \mathcal{A}(y)(w_N(y) - u_N(y)), (w_N(y) - u_N(y))\rangle \,\mathrm{d}\pi(y) \tag{5.36}$$

$$= \int_\Gamma \langle A_0^{-1/2} r_N(w_N)(y), A_0^{1/2}(w_N(y) - u_N(y))\rangle \,\mathrm{d}\pi(y) \tag{5.37}$$

$$\leq \|r_N\|_{A_0^{-1}} \|w_N - u_N\|_{A_0}, \tag{5.38}$$

where

$$\|w\|_{A_0^{-1}}^2 = \int_\Gamma \langle A_0^{-1} w(y), w(y)\rangle \,\mathrm{d}\pi(y) = \int_\Gamma \int_D a_0^{-1}(x)\nabla w(x,y)\cdot\nabla w(x,y)\,\mathrm{d}x\,\mathrm{d}\pi(y). \tag{5.39}$$

Because of the norm equivalence $\|w\|_{A_0} \sim \|w\|_{\mathcal{A}}$, we obtain the bound

$$\|w_N - u_N\|_{\mathcal{A}} \lesssim \|r_N\|_{A_0^{-1}} =: \iota(w_N). \tag{5.40}$$

Since $u_N, w_N \in \mathcal{V}_p(\mathcal{T};\Lambda)$, this is exactly

$$\|A_0^{-1/2} r_N\|_{A_0^{-1}} = \|\mathbf{A}_0^{-1/2}(\mathbf{A}(W) - F)\|_{\ell^2(\mathbb{R}^{N\times d_1\times\cdots\times d_M})} \tag{5.41}$$

in the Frobenius norm of the tensor space with $w_N = \sum_{i=0}^{N-1}\sum_{\mu\in\Lambda} W(i,\mu)\varphi_i P_\mu$.

6. Fully Adaptive Algorithm

The adaptive algorithm described in this section is similar to the algorithms presented in [20, 21] to which we refer for further details. In the following, we identify functional modules which encapsulate different aspects of the proposed adaptive approach. Given some mesh $\mathcal{T}$, a finite set $\Lambda \subset \mathcal{F}$ including 0, a fixed polynomial degree p and a tensor rank r, we assume that a numerical approximation $w_N \in \mathcal{V}_p(\Lambda,\mathcal{T})$ (ideally close to the Galerkin projection of (2.32)) is obtained by a function

$$w_N \leftarrow \mathtt{Solve}[\Lambda,\mathcal{T},r].$$

The error indicators of Section 5 and the overall upper bound $\eta_{\mathcal{A}}$ in Theorem 5.1 are computed by the methods

$$\begin{aligned}(\eta_T(w_N,\Lambda))_{T\in\mathcal{T}}, \eta(w_N,\Lambda,\mathcal{T}) &\leftarrow \mathtt{Estimate}_x[w_N,\Lambda,\mathcal{T}],\\ (\zeta_\nu(w_N))_{\nu\in\partial^\circ\Lambda}, \zeta(w_N,\partial\Lambda), (\|w_{N,\mu}\|_V)_{\mu\in\Lambda} &\leftarrow \mathtt{Estimate}_y[w_N,\Lambda],\\ \iota(w_N) &\leftarrow \mathtt{Estimate}_{\mathrm{ALS}}[w_N].\end{aligned}$$

With these, a separate marking of elements of the mesh $\mathcal{T}$ and the modes of the inactive boundary $\partial^\circ \Lambda \subset \mathcal{F} \setminus \Lambda$ of Λ is carried out by the functions

$$\mathcal{M} \leftarrow \texttt{Mark}_x[\vartheta_x, (\eta_T(w_N, \Lambda))_{T\in\mathcal{T}}, \eta(w_N, \Lambda, \mathcal{T})],$$
$$\Delta \leftarrow \texttt{Mark}_y[\vartheta_y, (\zeta_\nu(w_N))_{\nu\in\partial^\circ\Lambda}, \zeta(w_N, \partial\Lambda), (\|w_{N,\mu}\|_V)_{\mu\in\Lambda}]$$

with refinement parameters $0 < \vartheta_x, \vartheta_y \leq 1$. $\texttt{Refine}_x$ determines a subset $\mathcal{M} \subseteq \mathcal{T}$ of elements to refine. Details about the procedure $\texttt{Refine}_y$ can be found in [21, Sec. 5.2]. The obtained smallest sets $\mathcal{M} \subseteq \mathcal{T}$ and $\Delta \subset \partial\Lambda$ satisfy the Dörfler property, i.e.,

$$\eta(w_N, \Lambda, \mathcal{M}) \geq \vartheta_x \eta(w_N, \Lambda, \mathcal{T}) \quad \text{and} \quad \zeta(w_N, \Delta) \geq \vartheta_y \zeta(w_N, \partial\Lambda),$$

where $\eta(w_N, \Lambda, \mathcal{M})$ and $\zeta(w_N, \Delta)$ denote the obvious restrictions of the estimators η and ζ to the respective subsets. With these marking sets, the following methods produce a refined regular mesh $\mathcal{T}^*$ and an enlarged active set Λ^*, namely,

$$\mathcal{T}^* \leftarrow \texttt{Refine}_x[\mathcal{T}, \mathcal{M}] \quad \text{and} \quad \Lambda^* \leftarrow \texttt{Refine}_y[\Lambda, \Delta].$$

We set $\Lambda^* = \Lambda \cup \Delta$ but other choices are possible as long as we maintain a full tensor set.

In addition to the algorithm of previous publications, we also have to handle the case that the consistency error due to the inexact ALS solve in TT format estimated by $\iota(w_N)$ dominates. As described in Section 4.2, each rank of the tensor representation then is increased by one which is encoded in the routine

$$U^* \leftarrow \texttt{Update}_{\text{TT}}[U]$$

as outlined in Section 4.2

A single iteration step of an adaptive algorithm which returns either a refined $\mathcal{T}^*$ or Λ^* or the tensor format solution with increased rank U^* is given by the function $\texttt{TTASGFEM}$. For numerical simulations as performed in Section 7, this function has to be called iteratively until either a defined error bound or a maximum problem size is reached. The upper error bound directly follows from Theorem 5.1 and the computable estimators of Section 5.

$U^*, \mathcal{T}^*, \Lambda^* \leftarrow \texttt{TTASGFEM}[\Lambda, \mathcal{T}, \vartheta_x, \vartheta_y, r, \alpha_{\text{TT}}]$

$U^*, \mathcal{T}^*, \Lambda^* \leftarrow U, \mathcal{T}, \Lambda$
$w_N \leftarrow \texttt{Solve}[\Lambda, \mathcal{T}, r]$
$(\zeta_\nu)_{\nu\in\partial^\circ\Lambda}, \zeta, (\|w_{N,\mu}\|_V)_{\mu\in\Lambda} \leftarrow \texttt{Estimate}_y[w_N, \Lambda]$
$(\eta_T)_{T\in\mathcal{T}}, \eta \leftarrow \texttt{Estimate}_x[w_N, \Lambda, \mathcal{T}]$
$\iota \leftarrow \texttt{Estimate}_{\text{ALS}}[w_N]$
if $\eta = \max\{\eta, \zeta, \alpha_{TT}\iota\}$ **then**
  $\mathcal{M} \leftarrow \texttt{Mark}_x[\vartheta_x, (\eta_T)_{T\in\mathcal{T}}, \eta]$
  $\mathcal{T}^* \leftarrow \texttt{Refine}_x[\mathcal{T}, \mathcal{M}]$
else if $\zeta = \max\{\eta, \zeta, \alpha_{TT}\iota\}$ **then**
  $\Delta \leftarrow \texttt{Mark}_y[\vartheta_y, (\zeta_\nu)_{\nu\in\partial^*\Lambda}, \zeta, (\|w_{N,\mu}\|_V)_{\mu\in\Lambda}]$
  $\Lambda^* \leftarrow \texttt{Refine}_y[\Lambda, \Delta]$
else
  $U^* \leftarrow \texttt{Update}_{\text{TT}}[U]$

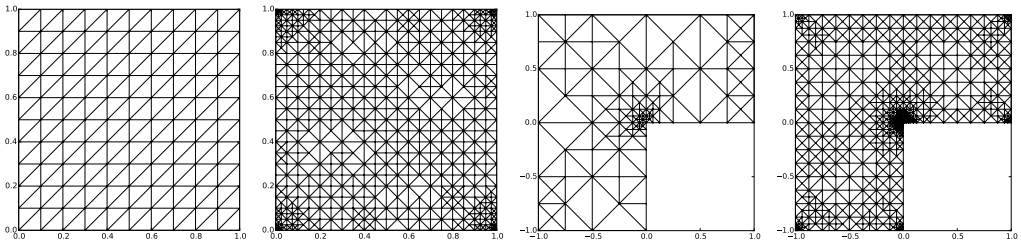

FIGURE 1. Adaptively refined meshes for square and L-shaped domains with $\tilde{\sigma} = 2$. Iterations 2 and 47 (square), and 12 and 43 (L-shaped).

7. NUMERICAL EXPERIMENTS

The adaptive algorithm of Section 6 is implemented with the open source framework `ALEA` [23] which has already been used for the ASGFEM presented in [20, 21], the `FEniCS` FE framework [28] and `ttpy` [52], an open source toolbox for the TT format. For the evaluation of the stochastic energy error of the numerical solution, we employ Monte Carlo sampling based on a realization-wise reference solution as described in Section 7.1. Subsequently, the performance of the new exact error estimator with regard to some benchmark problems on the square and the L-shaped domain as in [20, 21] is examined in Section 7.2.

7.1. **Evaluation of the error.** For experimental verification of the reliability of the error estimator, the error of the parametric solution is computed by Monte Carlo simulations. For this, a set of M_{MC} independent realizations $\{y^{(i)}\}_{i=1}^{M_{\mathrm{MC}}}$ of the stochastic parameters is determined. The $y_m^{(i)}$ are sampled according to the probability measure π of the (vector-valued) random variable $y \in \Gamma$. The mean-square error e of the (inexact) parametric SGFEM solution $w_N \in \mathcal{V}_p(\Lambda, \mathcal{T})$ is approximated by a Monte Carlo sample average

$$\|e\|_{\mathcal{X}}^2 = \int_\Gamma \|u(y) - w_N(y)\|_{\mathcal{X}}^2 \,\mathrm{d}\pi(y) \approx \frac{1}{M_{\mathrm{MC}}} \sum_{i=1}^{M_{\mathrm{MC}}} \|\tilde{u}(y^{(i)}) - w_N(y^{(i)})\|_{\mathcal{X}}^2 \,. \tag{7.1}$$

Note that the sampled solutions $\tilde{u}(y^{(i)})$ are approximations of the exact $u(y^{(i)}) = A^{-1}(y^{(i)})f$ since the differential operator is discretized on a fine reference mesh which is obtained by another uniform refinement of the adapted mesh generated from the SGFEM discretization of the final iteration. Moreover, the truncated expansion (2.8) of the random field $a(x, y)$ is expanded by the trailing largest 200 terms which are not considered by the best approximate parametric solution. We choose $M_{\mathrm{MC}} = 150$ for the Monte Carlo sampling of the reference error (7.1) which proved to be sufficient to assess the reliability of the error estimator.

7.2. **The stochastic diffusion problem.** We examine numerical simulations for the stationary diffusion problem (2.1) in a plane, polygonal domain $D \subset \mathbb{R}^2$. As in [20, 21], the expansion coefficients of the stochastic field (2.8) are given by

$$a_m(x) := \alpha_m \cos(2\pi \varrho_1(m) x_1) \cos(2\pi \varrho_2(m) x_2) \tag{7.2}$$

where α_m is of the form $\bar{\alpha} m^{-\tilde{\sigma}}$ with $\tilde{\sigma} > 1$ and some $0 < \bar{\alpha} < 1/\zeta(\tilde{\sigma})$ with the Riemann zeta function ζ. Then, (2.11) holds with $\gamma = \bar{\alpha}\zeta(\tilde{\sigma})$. Moreover,

$$\varrho_1(m) = m - k(m)(k(m)+1)/2 \quad \text{and} \quad \varrho_2(m) = k(m) - \varrho_1(m) \tag{7.3}$$

with $k(m) = \lfloor -1/2 + \sqrt{1/4 + 2m} \rfloor$, i.e., the coefficient functions a_m enumerate all planar Fourier sine modes in increasing total order. To illustrate the influence which the stochastic coefficient plays in the adaptive algorithm, we examine the expansion with slow and fast decay of α_m, setting $\tilde{\sigma}$ in (7.2) to either 2 or 4. The computations are carried out with conforming FEM spaces of polynomial degrees 1, 2 and 3. For the adaptive algorithm TTASGFEM of Section 6, the marking parameters are $\vartheta_x = \vartheta_y = 1/2$.

7.2.1. *Square domain.* The first example is the stationary diffusion equation (2.1) on the unit square $D = (0,1)^2$ with homogeneous Dirichlet boundary conditions and with right-hand side $f \equiv 1$. The results of the adaptive algorithm TTASGFEM for a slow decay of the coefficients with $\tilde{\sigma} = 2$ and a fast decay with $\tilde{\sigma} = 4$ are depicted in Figures 2–5. The amplitude $\bar{\alpha}$ in (7.2) was chosen as $\gamma/\zeta(\tilde{\sigma})$ with $\gamma = 0.9$, resulting in $\bar{\alpha} \approx 0.547$ for $\tilde{\sigma} = 2$ and $\bar{\alpha} \approx 0.832$ for $\tilde{\sigma} = 4$.

Figure 2 shows the error estimator $\eta_{\mathcal{A}}$ and the reference error obtained by Monte Carlo sampling as described in Section 7.1 in the top row. In all cases, the error estimator exhibits the same decay rate as the actual energy error. This is the largest for $p = 3$, as expected. In particular, the obtained accuracy for $p = 2, 3$ is significantly higher than for the first order FE approximation $p = 1$. With faster decay of the coefficient (right column), the attained error levels are better than for slower decay (left column). In the bottom row of the same figure, the progression of the maximal TT rank and the number of active stochastic dimensions (ydim) are pictured. It can be seen that an increase of stochastic dimensions is coupled to a growing rank of the TT representation of the solution. The largest values can be observed for $p = 3$ with slow decay, namely about 130 stochastic dimensions and the maximal TT rank 22. For the low-order $p = 1$ FEM, the stochastic dimensions and consequently the TT ranks are much lower since the deterministic approximation error is much more pronounced than for $p = 3$.

The compression level gained by the low-rank TT representation can be seen in Figure 3. There, the complete degrees of freedom of the problem discretization are plotted against the dofs of the compressed TT representation. Clearly, one can observe an exponential scaling with the number of active stochastic dimensions in the uncompressed case. Due to the curse of dimensionality, in case of slow decay and $p = 3$, the complete dofs rise to about 10^{50} which is far beyond a feasible problem size for actual computations. However, due to the employed tensor compression, the discretization is reduced significantly while retaining a very high approximation quality. This effect is less strongly pronounced for fewer active stochstastic dimensions as with the $p = 1$ discretizations.

The top row of Figure 4 depicts the deterministic FE dofs and the stochastic dofs of the discretization without the low-rank compression of the TT representation. The successive increase of active stochastic dimensions (plotted in Figure 2 bottom row) leads to an exponential growth of stochastic dofs, also compare with Figure 3.

In the TT representation, the increase of stochastic dofs basically becomes linear as pictured in the bottom row of Figure 4. It can also be observed that the successive rank increase leads to an increasing number of deterministic dofs. This is some negligible trade-off for the significant compression of the exponential growth of the stochastic dofs.

The degree of the Legendre chaos polynomials for the first 30 stochastic dimensions is depicted in Figure 5. As expected, for $p = 1$ the maximal polynomial degree is relatively small and only

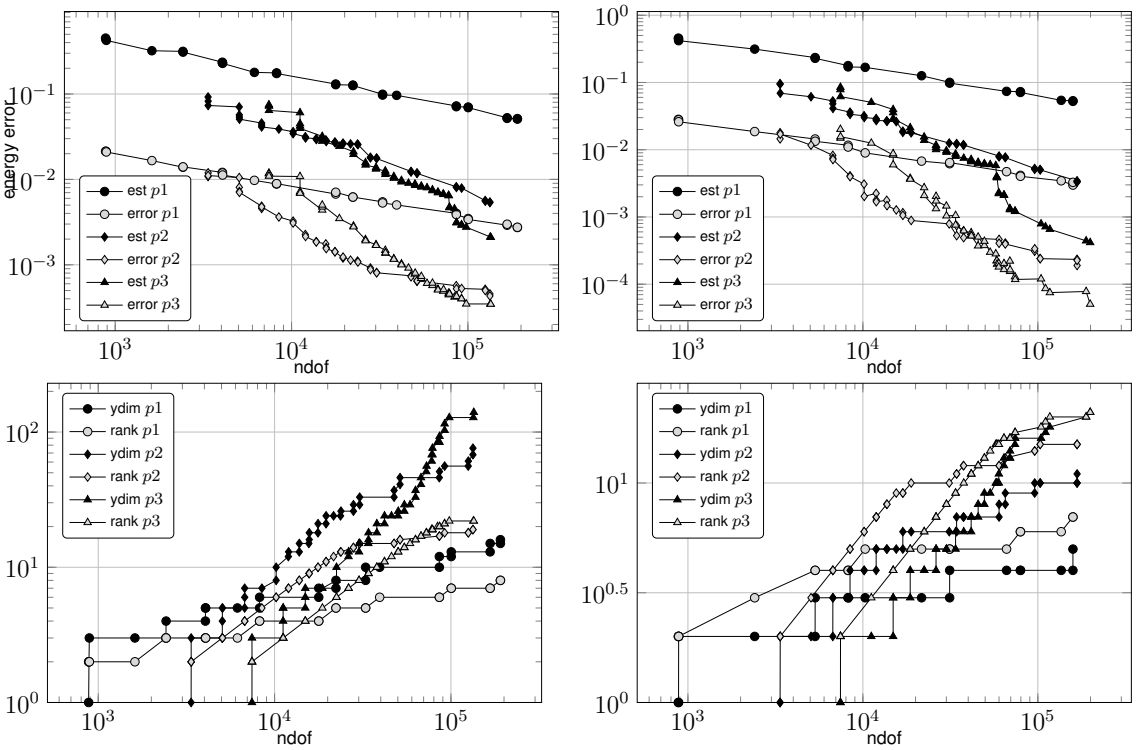

FIGURE 2. Convergence of the error estimator in the energy norm with FEM of degree $p = 1, 2, 3$ for the stationary diffusion problem on the square with homogeneous Dirichlet boundary conditions for slow ($\tilde{\sigma} = 2$, left) and fast ($\tilde{\sigma} = 4$, right) decay. Error estimator and sampled error (top), number of active stochastic dimensions and TT rank (bottom) for TT degrees of freedom.

few dimensions are active. Opposite to this, for $p = 3$ the stochastic error dominates and the stochastic dimensions are thus increased drastically. In case of fast decay, the first stochastic dimension is descretized with polynomials up to degree 10. This is smaller in case of slow decay, namely degree 5. However, the number of active dimensions then is about 130.

A comparison with the results obtained in [21] is depicted in Figure 6 for $p = 1, 3$. It can clearly be seen that the TT approximation presented in this work improves on previous results. For slow decay (left), the improvement is noticeable but the comparatively small stochastic basis selected in the adaptive algorithm of [21] already seems to be nearly optimal. For fast decay (right), significant improvements of the approximation quality can be observed.

7.2.2. *Square domain TT rank test.* In this example, we investigate the influence the TT rank has on the accuracy of the discrete solution. As in Section 7.2.1, we assume a square domain and determine the discrete solution u_N with $p = 1, 2, 3$ FEM, $M = 30$ stochastic dimensions and a slow decay rate $\tilde{\sigma} = 2$ in the coefficient representation. The stochastic variables are discretized with a uniform polynomial degree of 3. Figure 7 depicts the mean square energy error of the discrete solution subject to the rank of the low-rank TT representation. With increasing rank, the number of degrees of freedoms (dofs) continuously increased, starting with rank 1 and ending at about 10^5 dofs in the shown graphs. We compare the error progression with increasing ranks for each polynomial FE degree for a coarser and a finer grid. The left graph in Figure 7 depicts the

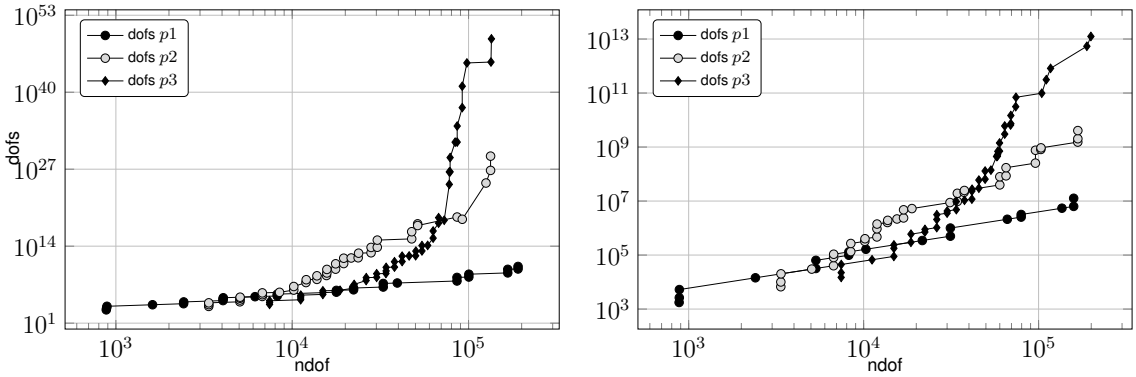

FIGURE 3. Progress of full degrees of freedom versus TT degrees of freedom for FEM of degree $p = 1, 2, 3$ for the stationary diffusion problem on the square with homogeneous Dirichlet boundary conditions for slow ($\tilde{\sigma} = 2$, left) and fast ($\tilde{\sigma} = 4$, right) decay.

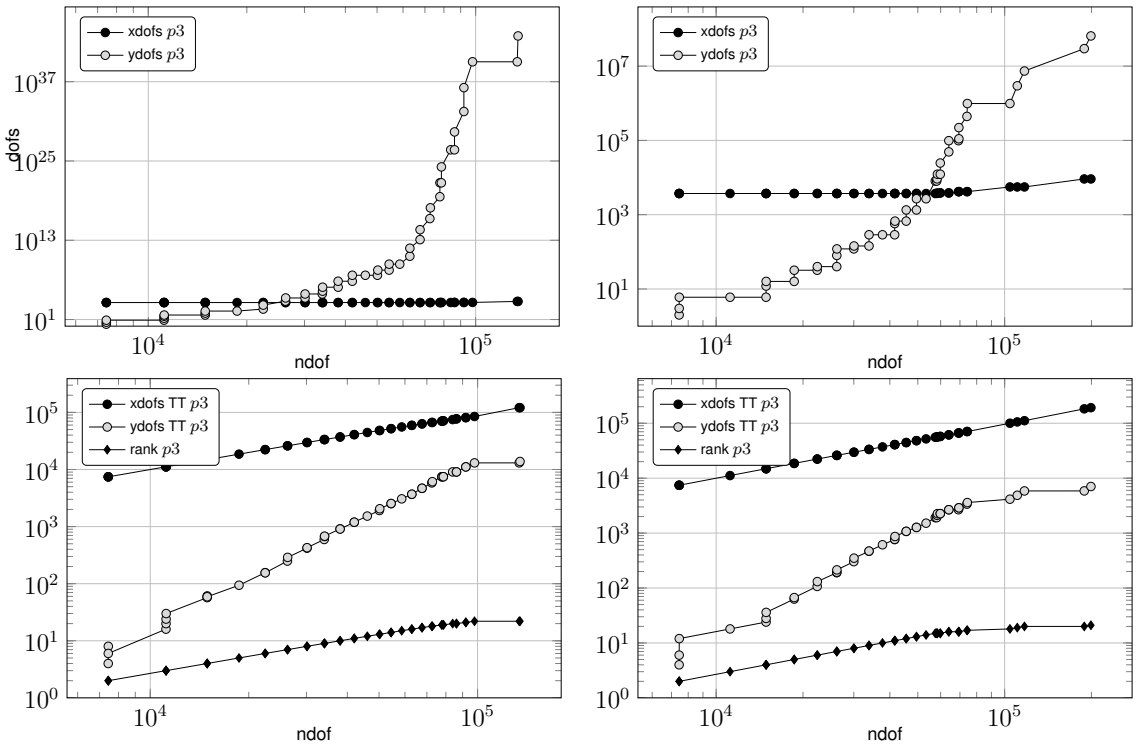

FIGURE 4. Degrees of freedom for deterministic FE (xdofs) and stochastic (ydofs) discretizations versus TT degrees of freedom for FE degree $p = 3$ for the stationary diffusion problem on the square with homogeneous Dirichlet boundary conditions for slow ($\tilde{\sigma} = 2$, left) and fast ($\tilde{\sigma} = 4$, right) decay. Uncompressed dofs (top row) and dofs in low-rank TT representation (xdofs TT and ydofs TT, bottom row) with TT ranks.

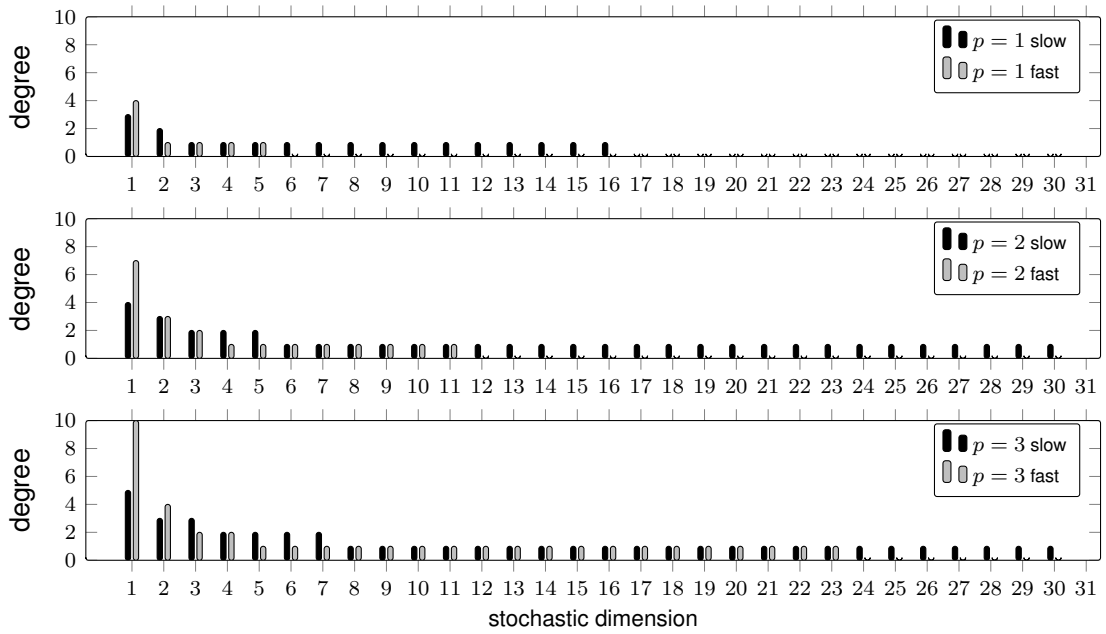

FIGURE 5. Polynomial degree for each stochastic mode with FEM of degree $p = 1, 2, 3$ (top to bottom) for the stationary diffusion problem on the square domain with homogeneous Dirichlet boundary conditions for slow ($\tilde{\sigma} = 2$) and fast ($\tilde{\sigma} = 4$) decay.

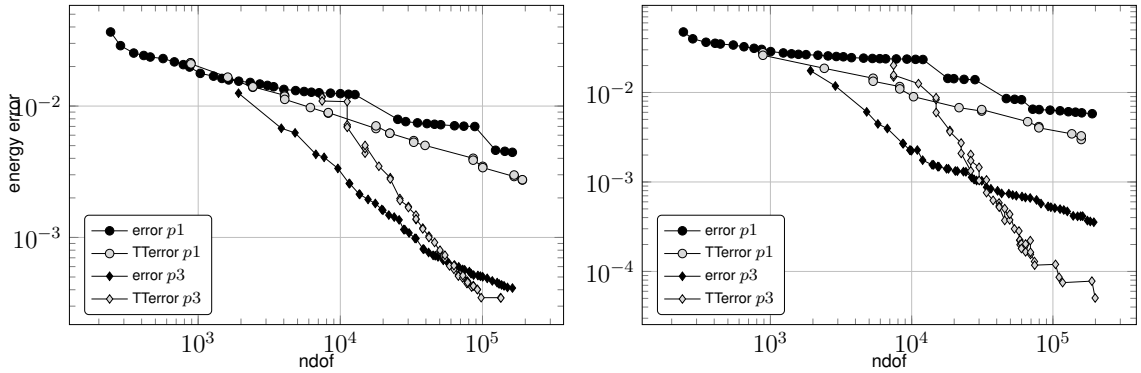

FIGURE 6. Comparison with results from [21] for sampled error in the energy norm with FEM of degree $p = 1, 3$ for the stationary diffusion problem on the square with homogeneous Dirichlet boundary conditions for slow ($\tilde{\sigma} = 2$, left) and fast ($\tilde{\sigma} = 4$, right) decay.

experiments with the coarser grid, the right graph shows the results on the finer grid. It can clearly be observed that the error crucially depends on the rank of the representation. With the coarser grid, a constant error decay can only be observed for $p = 3$. The solutions with $p = 1, 2$ show degraded error reduction for increasing ranks which is due to the prevailing FE approximation error which in these cases is dominant on the coarse mesh (left graph). This behavior changes on a finer grid (right graph). Since the FE approximation error is reduced in for all FE polynomial degrees, the error can be decreased for all $p = 1, 2, 3$. While $p = 2$ now exhibits a constant decay rate, $p = 1$ levels off once the FE error dominates again. This point is reached with rank 4.

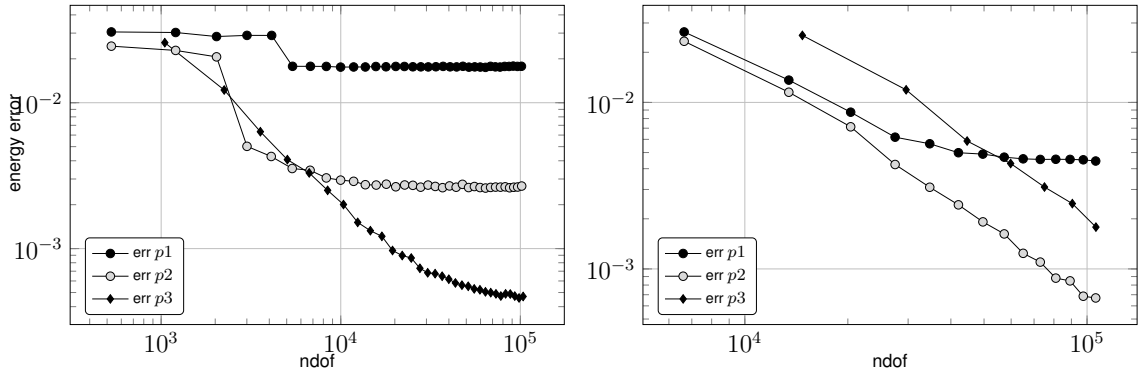

FIGURE 7. Convergence of energy error for sole rank increase of $p = 1, 2, 3$ solution on square domain with slow decay ($\tilde{\sigma} = 2$), isotropic stochastic degree 3 and $M = 30$ stochastic dimensions. Computation on coarse grid (left) and finer grid (right) with starting rank 1 and subsequent increase by 1 in each iteration step which leads to rank 12 in the final step.

A consequence we can infer from these observations is that, in order to obtain a fully adaptivity approximation scheme, also the tensor rank has to be updated according to the required accuracy of the numerical solution. Hence, this is included in the adaptive algorithm presented in Section 6.

7.2.3. *L-shaped domain.* A standard benchmark problem for deterministic a posteriori error estimators is the stationary diffusion problem (2.1) on the L-shaped domain $D = (-1, 1)^2 \setminus (0, 1) \times (-1, 0)$. It is well-known that the solution exhibits a singularity at the reentrant corner at $(0, 0)$ which has to be resolved by a pronounced mesh refinement in its vicinity in order to achieve optimal convergence rates, also see Figure 1. The previous remarks in Section 7.2.1 regarding the setup of the coefficient and the error evaluation are also valid with this example. The convergence of the error estimator and its efficiency with regard to the error determined by (7.1) are depicted in Figures 8–9.

Similar to the example on the square domain of Section 7.2.1, the energy error and the a posteriori error estimator for slow (left) and fast (right) decay of the stochastic field coefficient are depicted in Figure 8 (top row). Moreover, the number of active stochastic dimensions and the TT rank of the tensor representation are shown (bottom row). Compared to the experiments on the square domain, the stochastic discretization reaches somewhat lower levels since the adaptive algorithm first has to ensure that the determistic FE approximation is sufficiently accurate due to the present corner singularity. In particular, this results in an initial mesh refinement for $p = 2, 3$ which was not the case before. Hence, the number of active stochastic dimensions now stays below 100 in all experiments.

For different FE degrees, the observed convergence rates of the energy error are obviously different with $p = 3$ leading to the highest rates.

The distribution of stochastic polynomial degrees with respect to the stochastic dimension is pictured in Figure 9. These are similar to Figure 5 with somewhat lower degrees than before due to the reasons mentioned above.

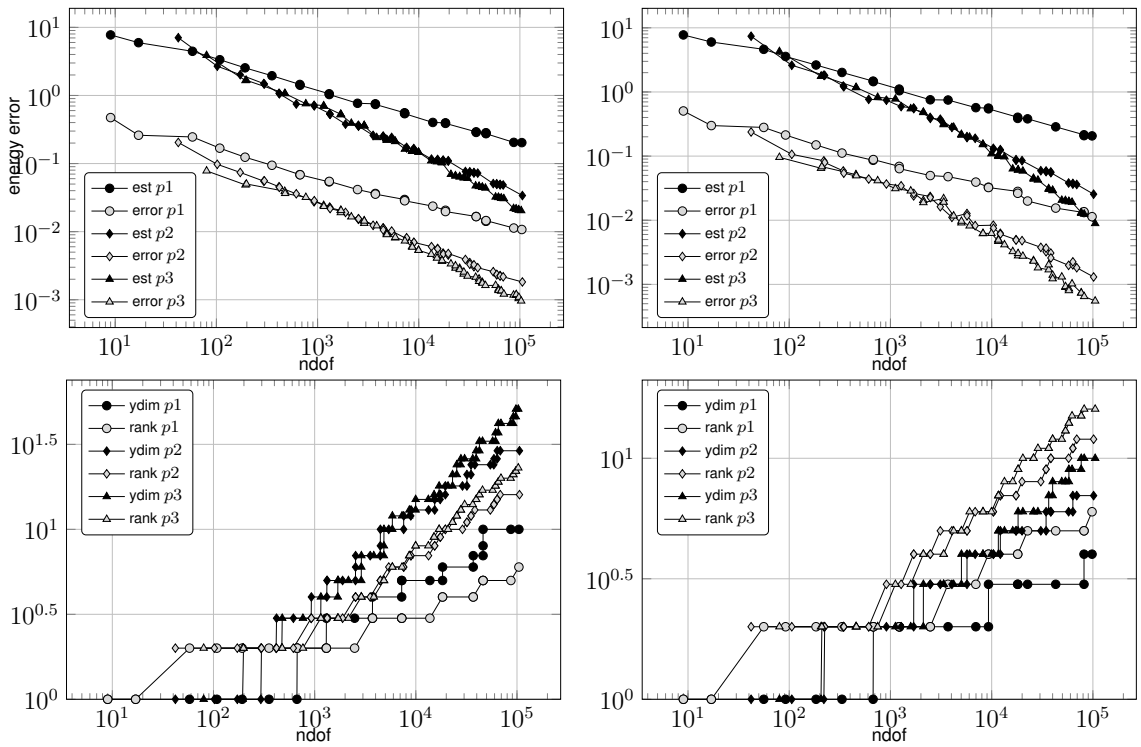

FIGURE 8. Convergence of the error estimator in the energy norm with FEM of degree $p = 1, 2, 3$ for the stationary diffusion problem on the L-shaped domain with homogeneous Dirichlet boundary conditions for slow ($\tilde{\sigma} = 2$, left) and fast ($\tilde{\sigma} = 4$, right) decay. Error estimator and sampled error (top), number of active stochastic dimensions and TT rank (bottom) for TT degrees of freedom.

REFERENCES

[1] Mark Ainsworth and J. Tinsley Oden. *A posteriori error estimation in finite element analysis*. Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], New York, 2000.

[2] Ivo Babuška and Panagiotis Chatzipantelidis. On solving elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 191(37-38):4093–4122, 2002.

[3] Ivo Babuška, Fabio Nobile, and Raúl Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, 45(3):1005–1034, 2007.

[4] Ivo Babuška, Raúl Tempone, and Georgios E. Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J. Numer. Anal.*, 42(2):800–825, 2004.

[5] Ivo Babuška, Raúl Tempone, and Georgios E. Zouraris. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Comput. Methods Appl. Mech. Engrg.*, 194(12-16):1251–1294, 2005.

[6] Jonas Ballani and Lars Grasedyck. Hierarchical tensor approximation of output quantities of parameter-dependent pdes. *Preprint*, 385, 2014.

[7] Alex Bespalov, Catherine E. Powell, and David Silvester. Energy norm a posteriori error estimation for parametric operator equations. *SIAM J. Sci. Comput.*, 36(2):A339–A363, 2014.

[8] Dietrich Braess. *Finite elements*. Cambridge University Press, Cambridge, third edition, 2007. Theory, fast solvers, and applications in elasticity theory, Translated from the German by Larry L. Schumaker.

[9] Carsten Carstensen, Martin Eigel, Ronald H.W. Hoppe, and Caroline Löbhard. A review of unified a posteriori finite element error control. *Numer. Math. Theor. Meth. Appl.*, 5(4):509–558, 2012.

[10] Peng Chen, Alfio Quarteroni, and Gianluigi Rozza. A weighted reduced basis method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 51(6):3163–3185, 2013.

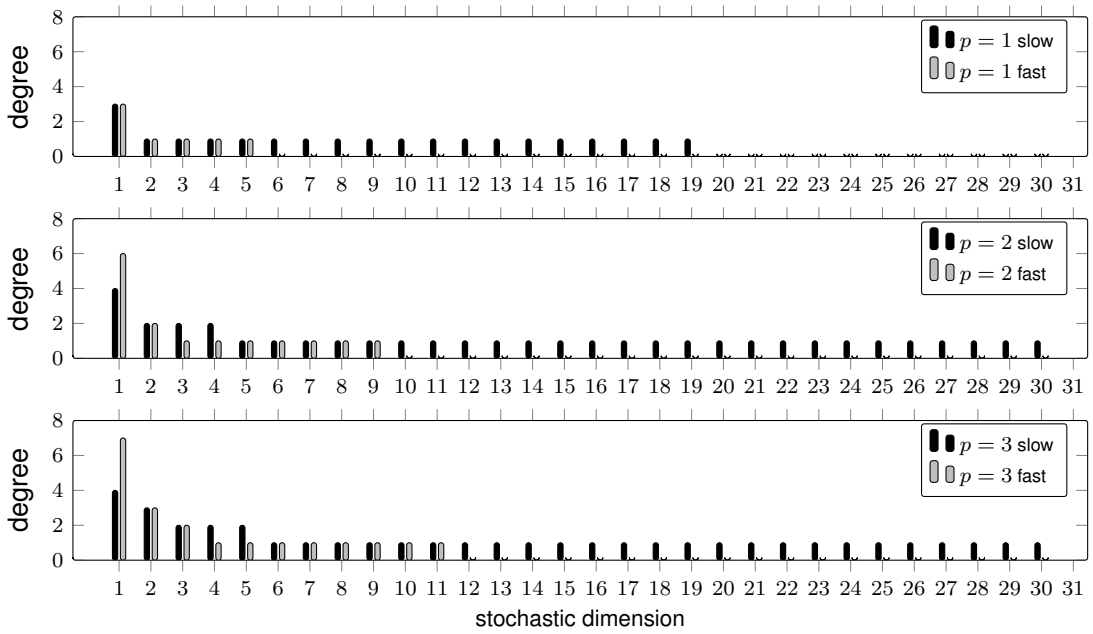

FIGURE 9. Polynomial degree for each stochastic mode with FEM of degree $p = 1, 2, 3$ (top to bottom) for the stationary diffusion problem on the L-shaped domain with homogeneous Dirichlet boundary conditions for slow ($\tilde{\sigma} = 2$) and fast ($\tilde{\sigma} = 4$) decay.

[11] Peng Chen, Alfio Quarteroni, and Gianluigi Rozza. Comparison between reduced basis and stochastic collocation methods for elliptic problems. *J. Sci. Comput.*, 59(1):187–216, 2014.

[12] Albert Cohen, Ronald DeVore, and Christoph Schwab. Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs. *Found. Comput. Math.*, 10(6):615–646, 2010.

[13] Albert Cohen, Ronald Devore, and Christoph Schwab. Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's. *Anal. Appl. (Singap.)*, 9(1):11–47, 2011.

[14] Vin de Silva and Lek-Heng Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix Anal. Appl.*, 30(3):1084–1127, September 2008.

[15] Manas K. Deb, Ivo M. Babuška, and J. Tinsley Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Comput. Methods Appl. Mech. Engrg.*, 190(48):6359–6372, 2001.

[16] Sergey Dolgov, Boris N. Khoromskij, Alexander Litvinenko, and Hermann G. Matthies. Computation of the response surface in the tensor train data format. Submitted / accepted., 2014.

[17] Sergey Dolgov, Boris N Khoromskij, Alexander Litvinenko, and Hermann G Matthies. Computation of the response surface in the tensor train data format. *arXiv preprint arXiv:1406.2816*, 2014.

[18] Sergey Dolgov, Boris N Khoromskij, Alexander Litvinenko, and Hermann G Matthies. Polynomial chaos expansion of random coefficients and the solution of stochastic partial differential equations in the tensor train format. *arXiv preprint arXiv:1503.03210*, 2015.

[19] Sergey V Dolgov and Dmitry V Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM Journal on Scientific Computing*, 36(5):A2248–A2271, 2014.

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

[21] 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.

[22] Martin Eigel and Christian Merdon. Local equilibration error estimators for guaranteed error control in adaptive stochastic higher-order galerkin fem. *WIAS Preprint*, (1997), 2014.

[23] Martin Eigel and Elmar Zander. `alea` - A Python Framework for Spectral Methods and Low-Rank Approximations in Uncertainty Quantification, `https://bitbucket.org/aleadev/alea`.

[24] Oliver G. Ernst, Antje Mugler, Hans-Jörg Starkloff, and Elisabeth Ullmann. On the convergence of generalized polynomial chaos expansions. Technical Report 60, DFG Schwerpunktprogramm 1324, 2010.

[25] Mike Espig, Wolfgang Hackbusch, and Aram Khachatryan. On the Convergence of Alternating Least Squares Optimisation in Tensor Format Representations. 423, 2015.

[26] Mike Espig, Wolfgang Hackbusch, Alexander Litvinenko, Hermann G. Matthies, and Philipp Wähnert. Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats. Preprint, Max Planck Institute for Mathematics in the Sciences, 2012.

[27] Mike Espig, Wolfgang Hackbusch, Alexander Litvinenko, Hermann G Matthies, and Elmar Zander. Efficient analysis of high dimensional data in tensor formats. In *Sparse Grids and Applications*, pages 31–56. Springer, 2013.

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

[29] Philipp Frauenfelder, Christoph Schwab, and Radu Alexandru Todor. Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):205–228, 2005.

[30] Luis David Garcia, Michael Stillman, and Bernd Sturmfels. Algebraic geometry of bayesian networks. *Journal of Symbolic Computation*, 39(3âĂŞ4):331 – 355, 2005. Special issue on the occasion of {MEGA} 2003.

[31] Roger G Ghanem and Robert M Kruger. Numerical solution of spectral stochastic finite element systems. *Computer Methods in Applied Mechanics and Engineering*, 129(3):289–303, 1996.

[32] Roger G. Ghanem and Pol D. Spanos. *Stochastic finite elements: a spectral approach*. Springer-Verlag, New York, 1991.

[33] Claude Jeffrey Gittelson. Stochastic Galerkin approximation of operator equations with infinite dimensional noise. Technical Report 2011-10, Seminar for Applied Mathematics, ETH Zürich, 2011.

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

[35] Max D. Gunzburger, Clayton G. Webster, and Guannan Zhang. Stochastic finite element methods for partial differential equations with random input data. *Acta Numer.*, 23:521–650, 2014.

[36] Wolfgang Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2012.

[37] Wolfgang Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42 of *Springer series in computational mathematics*. Springer, Heidelberg, 2012.

[38] Wolfgang Hackbusch. Numerical tensor calculus. *Acta Numer.*, 23:651–742, 2014.

[39] Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, 2009.

[40] Wolfgang Hackbusch and Reinhold Schneider. Tensor spaces and hierarchical tensor representations. In *Extraction of Quantifiable Information from Complex Systems*, pages 237–261. Springer, 2014.

[41] Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. On manifolds of tensors of fixed TT-rank. *Numerische Mathematik*, 120(4):701–731, 2012.

[42] Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. On manifolds of tensors of fixed TT-rank. *Numerische Mathematik*, 120(4):701–731, 2012.

[43] Boris N. Khoromskij and Ivan V. Oseledets. Quantics-TT collocation approximation of parameter-dependent and stochastic elliptic PDEs. *Comput. Methods Appl. Math.*, 10(4):376–394, 2010.

[44] Boris N. Khoromskij and Christoph Schwab. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. *SIAM J. Sci. Comput.*, 33(1):364–385, 2011.

[45] Boris N. Khoromskij and Christoph Schwab. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. *SIAM journal on scientific computing*, 33(1):364–385, 2011.

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

[47] Joseph M. Landsberg. *Tensors: Geometry and Applications*. Graduate studies in mathematics. American Mathematical Society, 2012.

[48] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, March 2000.

[49] Hermann G. Matthies and Andreas Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 194(12-16):1295–1331, 2005.

[50] Fabio Nobile, Raul Tempone, and Clayton G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.

[51] Fabio Nobile, Raul Tempone, and Clayton G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2309–2345, 2008.

[52] Ivan V. Oseledets. ttpy - A Python Implementation of the TT-Toolbox, `https://github.com/oseledets/ttpy`.

[53] Ivan V. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011.

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

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

[56] Manuel F Pellissetti and Roger G Ghanem. Iterative solution of systems of linear equations arising in the context of stochastic finite elements. *Advances in Engineering Software*, 31(8):607–616, 2000.

[57] Catherine E. Powell and Howard C. Elman. Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA Journal of Numerical Analysis*, 29:350–375, 2009.

[58] T. Rohwedder and A. Uschmajew. On local convergence of alternating schemes for optimization of convex problems in the tensor train format. *SIAM Journal on Numerical Analysis*, 51(2):1134–1162, 2013.

[59] Erhard Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. I. Teil: Entwicklung willkürlicher Funktionen nach Systemen vorgeschriebener. *Mathematische Annalen*, 63:433–476, 1907.

[60] Reinhold Schneider and André Uschmajew. Convergence results for projected line-search methods on varieties of low-rank matrices via Łojasiewicz inequality. *SIAM J. Optim.*, 25(1):622–646, 2015.

[61] Christoph Schwab and Claude Jeffrey Gittelson. Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs. *Acta Numer.*, 20:291–467, 2011.

[62] Szilárd Szalay, Max Pfeffer, Valentin Murg, Gergely Barcza, Frank Verstraete, Reinhold Schneider, and Örs Legeza. Tensor product methods and entanglement optimization for ab initio quantum chemistry. *International Journal of Quantum Chemistry*, pages n/a–n/a, 2015.

[63] Elisabeth Ullmann. A kronecker product preconditioner for stochastic galerkin finite element discretizations. *SIAM Journal on Scientific Computing*, 32(2):923–946, 2010.

[64] André Uschmajew and Bart Vandereycken. The geometry of algorithms using hierarchical tensors. *Linear Algebra and its Applications*, 439(1):133 – 166, 2013.

[65] R. Verfürth. *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Teubner Verlag and J. Wiley, Stuttgart, 1996.

[66] Dongbin Xiu and Jan S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27(3):1118–1139 (electronic), 2005.

[67] Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Comput. Methods Appl. Mech. Engrg.*, 191(43):4927–4948, 2002.