**Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig**

Efficient Analysis of High Dimensional Data in Tensor Formats

(revised version: October 2011)

by

*Mike Espig, Wolfgang Hackbusch, Alexander Litvinenko,
Hermann G. Matthies, and Elmar Zander*

Preprint no.: 62 2011

Efficient Analysis of High Dimensional Data in Tensor Formats

Mike Espig[1], Wolfgang Hackbusch[1], Alexander Litvinenko[2], Hermann G. Matthies[2] and Elmar Zander[2]

[1] Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany mike.espig@mis.mpg.de
[2] Technical University Braunschweig, Germany wire@tu-bs.de

In this article we introduce new methods for the analysis of high dimensional data in tensor formats, where the underling data come from the stochastic elliptic boundary value problem. After discretisation of the deterministic operator as well as the presented random fields via KLE and PCE, the obtained high dimensional operator can be approximated via sums of elementary tensors. This tensors representation can be effectively used for computing different values of interest, such as maximum norm, level sets and cumulative distribution function. The basic concept of the data analysis in high dimensions is discussed on tensors represented in the canonical format, however the approach can be easily used in other tensor formats. As an intermediate step we describe efficient iterative algorithms for computing the characteristic and sign functions as well as pointwise inverse in the canonical tensor format. Since during majority of algebraic operations as well as during iteration steps the representation rank grows up, we use lower-rank approximation and inexact recursive iteration schemes.

1 Introduction

Let us give an example which motivates much of the following formulation and development. Assume that we are interested in the time evolution of some system, described by

$$\frac{\mathrm{d}}{\mathrm{d}t}u(t) = A(p)(u(t)), \tag{1}$$

where $u(t)$ is in some Hilbert space $\mathcal{U}$ and $A(p)$ is some parameter dependent operator; in particular $A(p)$ could be some parameter-dependent differential operator, for example

$$\frac{\partial}{\partial t}u(x,t) = \nabla \cdot (\kappa(x,\omega)\nabla u(x,t)) + f(x,t), \quad x \in \mathcal{G} \subset \mathbb{R}^d, t \in [0,T] \tag{2}$$

where $\kappa(x,\omega)$ is a random field dependent on a random parameter in some probability space $\omega \in \Omega$, and one may take $\mathcal{U} = L_2(\mathcal{G})$.

One may for each $\omega \in \Omega$ seek for solutions in $L_2([0,T],\mathcal{U}) \cong L_2([0,T]) \otimes \mathcal{U}$. Assigning

$$\mathcal{S} = L_2([0,T]) \otimes L_2(\Omega),$$

one is looking for a solution in $\mathcal{U} \otimes \mathcal{S}$. $L_2(\Omega)$ can for random fields be further decomposed

$$L_2(\Omega) = L_2(\times_j \Omega_j) \cong \bigotimes_j L_2(\Omega_j) \cong \bigotimes_j L_2(\mathbb{R},\Gamma_j).$$

with some measures Γ_j. Then the parametric solution is sought in the space

$$\mathcal{U} \otimes \mathcal{S} = L_2(\mathcal{G}) \otimes \left(L_2([0,T]) \otimes \bigotimes_j L_2(\mathbb{R}, \Gamma_j) \right). \tag{3}$$

The more tensor factors there are, the more difficult and high-dimensional the problem will be. But on the other hand a high number of tensor factors in Eq. (3) will also allow very sparse representation and highly effective algorithms—this is of course assuming that the solution is intrinsically on a low-dimensional manifold and we 'just' need to discover it.

This paper is about exploiting the tensor product structure which appears in Eq. (3) for efficient calculations to be performed on the solution. This tensor product structure—in this case multiple tensor product structure—is typical for such parametric problems. What is often desired, is a representation which allows for the approximate evaluation of the state of Eq. (1) or Eq. (2) without actually solving the system again. Sometimes this is called a 'response surface'. Furthermore, one would like this representation to be inexpensive to evaluate, and for it to be convenient for certain post-processing tasks, for example like finding the minimum or maximum value over some or all parameter values.

1.1 Tensorial quantities

Computations usually require that one chooses finite dimensional subspaces and bases in there, in the example case of Eq. (2) these are

$$\begin{aligned} \text{span } \{w_n\}_{n=1}^N = \mathcal{U}_N \subset \mathcal{U}, &\quad \dim \mathcal{U}_N = N, \\ \text{span } \{\tau_k\}_{k=1}^K = \mathcal{T}_K \subset L_2([0,T]) = \mathcal{S}_I, &\quad \dim \mathcal{T}_K = K, \\ \forall m = 1, \dots, M : & \\ \text{span } \{X_{j_m}\}_{j_m=1}^{J_m} = \mathcal{S}_{II,J_m} \subset L_2(\mathbb{R}, \Gamma_m) = \mathcal{S}_{II}, &\quad \dim \mathcal{S}_{II,J_m} = J_m. \end{aligned}$$

Let $\mathcal{P} := [0,T] \times \Omega$, an approximation to $u : \mathcal{P} \to \mathcal{U}$ is thus given by

$$\begin{aligned} &u(x,t,\omega_1,\dots,\omega_M) \approx \\ &\qquad \sum_{n=1}^N \sum_{k=1}^K \sum_{j_1=1}^{J_1} \cdots \sum_{j_M=1}^{J_M} \hat{u}_{n,k}^{j_1,\dots,j_M} w^n(x) \otimes \tau^k(t) \otimes \left(\bigotimes_{m=1}^M X_{j_m}(\omega_m) \right). \end{aligned} \tag{4}$$

Via Eq. (4) the tensor $\hat{u}_{n,k}^{j_1,\dots,j_M}$ represents the state $u(x,t,\omega_1,\dots,\omega_M)$ and is thus a concrete example of a 'response surface'.

To allow easier interpretation later, assume that $\{x_1,\dots,x_N\} \subset \mathcal{G}$ are unisolvent points for $\{w_n\}_{n=1}^N$, and similarly $\{t_1,\dots,t_K\} \subset [0,T]$ are unisolvent points for $\{\tau_k\}_{k=1}^K$, and for each $m = 1,\dots,M$ the points $\{\omega_m^1,\dots,\omega_M^{J_m}\} \subset \Omega_m$ are unisolvent points for $\{X_{j_m}\}_{j_m=1}^{J_m}$. Then the same information which is in Eq. (4) is also contained in the evaluation at those unisolvent points:

$$\begin{aligned} &\forall n = 1,\dots,N,\ k = 1,\dots,K,\ m = 1,\dots,M,\ j_m = 1,\dots,J_m : \\ &\qquad u_{n,k}^{j_1,\dots,j_m,\dots,j_M} = u(x_n, t_k, \omega_1^{j_1}, \dots, \omega_m^{j_m}, \dots, \omega_M^{J_M}), \end{aligned} \tag{5}$$

this is just a different choice of basis for the tensor. In keeping with symbolic index notation, we denote by $(u_{n,k}^{j_1,\dots,j_m,\dots,j_M})$ the whole tensor in Eq. (5).

Model reduction or sparse representation may be applied before, during, or after the computation of the solution to Eq. (1) for new values of t or $(\omega_1,\dots,\omega_M)$. It may be performed in a pure Galerkin fashion by choosing even smaller, but well adapted subspaces, say for example $\mathcal{U}_{N'} \subset \mathcal{U}_N$, and thus reducing the dimensionality and hopefully also the work involved in a new solution. This is sometimes termed 'flat' Galerkin. In this kind of reduction, the subspace $\mathcal{U}_{N''} = \mathcal{U}_N \ominus \mathcal{U}_{N'}$ is completely neglected.

In nonlinear Galerkin methods, the part $u_{N'} \in \mathcal{U}_{N'}$ is complemented by a possibly non-linear map $v : \mathcal{U}_{N'} \to \mathcal{U}_{N''}$ to $u_N \approx u_{N'} + v(u_{N'}) \in \mathcal{U}_{N'} \oplus \mathcal{U}_{N''} = \mathcal{U}_N$. The approximate solution is not in a flat subspace anymore, but in some possibly non-linear manifold, hence the name. Obviously this procedure may be applied to any of the approximating subspaces.

Another kind of reduction works directly with the tensor $(u_{n,k}^{j_1,\dots,j_M})$ in Eq. (5). It has formally $R^{''} = N \times K \times \prod_{m=1}^{M} J_m$ terms. The minimum number R of terms needed to represent the sum is defined as the rank of that tensor. One might try to approximately express the sum with even fewer $R^{'} \ll R \leq R^{''}$ terms, this is termed a low-rank approximation. It may be seen as a non-linear model reduction.

In this way the quantity in Eq. (5) is expressed as

$$(u_{n,k}^{j_1,\dots,j_m,\dots,j_M}) \approx \sum_{\rho=1}^{R^{'}} u^{\rho} w_{\rho} \otimes \tau_{\rho} \otimes \left(\bigotimes_{m=1}^{M} X_{\rho_m} \right), \tag{6}$$

where $w_\rho \in \mathbb{R}^N$, $\tau_\rho \in \mathbb{R}^K$, and for each $m = 1, \dots, M$: $X_{\rho_m} \in \mathbb{R}^{J_m}$.

Hence Eq. (6) is an approximation for the response, another—sparse—'response surface'. With such a representation, one wants to perform numerous tasks, among them

- evaluation for specific parameters $(t, \omega_1, \dots, \omega_M)$,
- finding maxima and minima,
- finding 'level sets'.

2 Discretisation of diffusion problem with uncertain coefficient

Since the time dependence in Eq. (1) doesn't influence on the proposed further methods we demonstrate our theoretical and numerical results on the following stationary example

$$\begin{aligned} -\operatorname{div}(\kappa(x,\omega)\nabla u(x,\omega)) &= f(x,\omega) \quad \text{a.e. } x \in \mathcal{G}, \quad \mathcal{G} \subset \mathbb{R}^2, \\ u(x,\omega) &= 0 \qquad\qquad \text{a.e. } x \in \partial\mathcal{G}. \end{aligned} \tag{7}$$

This is a stationary diffusion equation described by a conductivity parameter $\kappa(x,\omega)$. It may, for example, describe the groundwater flow through a porous subsurface rock / sand formation [5, 16, 21, 30, 36]. Since the conductivity parameter in such cases is poorly known, i.e. it may be considered as uncertain, one may model it as a random field.

Let us introduce a bounded spatial domain of interest $\mathcal{G} \subset \mathbb{R}^d$ together with the hydraulic head u appearing in *Darcy's* law for the seepage flow $q = -\kappa\nabla u$, and f as flow sinks and sources. For the sake of simplicity we only consider a scalar conductivity, although a conductivity tensor would be more appropriate. The conductivity κ and the source f are defined as random fields over the probability space Ω. By introduction of this stochastic model of uncertainties Eq. (7) is required to hold almost surely in ω, i.e. $\mathbb{P}$-almost everywhere.

As the conductivity κ has to be positive, and is thus restricted to a particular case in a vector space, we consider its logarithm as the primary quantity, which may have any value. We assume that it has finite variance and thus choose for maximum entropy a Gaussian distribution. Hence the conductivity is initially log-normally distributed. Such kind of assumption is known as *a priori* information/distribution:

$$\kappa(x) := \exp(q(x)), \quad q(x) \sim N(0, \sigma_q^2). \tag{8}$$

In order to solve the stochastic forward problem we assume that $q(x)$ has covariance function of the exponential type $\mathrm{Cov}_q(x,y) = \sigma_q^2 \exp(-|x-y|/l_c)$ with prescribed covariance length l_c.

In order to make sure that the numerical methods will work well, we strive to have similar overall properties of the stochastic system Eq. (7) as in the deterministic case (for fixed ω). For this to hold, it is necessary that the operator implicitly described by Eq. (7) is continuous and continuously invertible, i.e. we require that both $\kappa(x,\omega)$ and $1/\kappa(x,\omega)$ are essentially bounded (have finite L_∞ norm) [2, 30, 27, 33]:

$$\kappa(x,\omega) > 0 \quad \text{a.e.}, \quad \|\kappa\|_{L_\infty(\mathcal{G}\times\Omega)} < \infty, \quad \|1/\kappa\|_{L_\infty(\mathcal{G}\times\Omega)} < \infty. \tag{9}$$

Two remarks are in order here: one is that for a heterogeneous medium each realisation $\kappa(x,\omega)$ should be modelled as a tensor field. This would entail a bit more cumbersome notation and not help to explain the procedure any better. Hence for the sake of simplicity we stay with the unrealistically simple model of a scalar conductivity field. The strong form given in Eq. (7) is not a good starting point for the Galerkin approach. Thus, as in the purely deterministic case, a variational formulation is needed, leading—via the *Lax-Milgram* lemma—to a well-posed problem. Hence, we search for $u \in \mathscr{U} := \mathcal{U} \otimes \mathcal{S}$ such that for all $v \in \mathscr{U}$ holds:

$$\mathbf{a}(v,u) := \mathbb{E}\left(\mathsf{a}(\omega)(v(\cdot,\omega), u(\cdot,\omega))\right) = \mathbb{E}\left(\langle \ell(\omega), v(\cdot,\omega)\rangle\right) =: \langle\!\langle \boldsymbol{\ell}, v\rangle\!\rangle. \tag{10}$$

Here $\mathbb{E}(b) := \mathbb{E}(b(\omega)) := \int_\Omega b(\omega)\, \mathbb{P}(\mathrm{d}\omega)$ is the expected value of the random variable (RV) b. The double bracket $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathscr{U}}$ is interpreted as duality pairing between $\mathscr{U}$ and its dual space $\mathscr{U}^*$.

The bi-linear form $\mathbf{a}$ in Eq. (10) is defined using the usual deterministic bi-linear (though parameter-dependent) form :

$$\mathsf{a}(\omega)(v,u) := \int_{\mathcal{G}} \nabla v(x) \cdot (\kappa(x,\omega) \nabla u(x))\ \mathrm{d}x, \tag{11}$$

for all $u, v \in \mathcal{U} := \mathring{H}^1(\mathcal{G}) = \{u \in H^1(\mathcal{G}) \mid u = 0 \text{ on } \partial\mathcal{G}\}$. The linear form $\boldsymbol{\ell}$ in Eq. (10) is similarly defined through its deterministic but parameter-dependent counterpart:

$$\langle \ell(\omega), v\rangle := \int_{\mathcal{G}} v(x) f(x,\omega)\ \mathrm{d}x, \quad \forall v \in \mathcal{U}, \tag{12}$$

where f has to be chosen such that $\ell(\omega)$ is continuous on $\mathcal{U}$ and the linear form $\boldsymbol{\ell}$ is continuous on $\mathscr{U}$, the Hilbert space tensor product of $\mathcal{U}$ and $\mathcal{S}$.

Let us remark that—loosely speaking—the stochastic weak formulation is just the expected value of its deterministic counterpart, formulated on the Hilbert tensor product space $\mathcal{U} \otimes \mathcal{S}$, i.e. the space of $\mathcal{U}$-valued RVs with finite variance, which is isomorphic to $L_2(\Omega, \mathbb{P}; \mathcal{U})$. In this way the stochastic problem can have the same theoretical properties as the underlying deterministic one, which is highly desirable for any further numerical approximation.

2.1 Spatial Discretisation

Let us discretise the spatial part of Eq. (10) by a standard finite element method. However, any other type of discretisation technique may be used with the same success. Since we deal with Galerkin methods in the stochastic space, assuming this also in the spatial domain gives the more compact representation of the problem. Let us take a finite element ansatz $\mathcal{U}_N := \{\varphi_n(x)\}_{n=1}^N \subset \mathcal{U}$ [34, 6, 39] as a corresponding subspace, such that the solution may be approximated by:

$$u(x,\omega) = \sum_{n=1}^{N} u_n(\omega)\varphi_n(x), \tag{13}$$

where the coefficients $\{u_n(\omega)\}$ are now RVs in $\mathcal{S}$. Inserting the ansatz Eq. (13) back into Eq. (10) and applying the spatial Galerkin conditions [30, 27], we arrive at:

$$\boldsymbol{A}(\omega)[\boldsymbol{u}(\omega)] = \boldsymbol{f}(\omega), \tag{14}$$

where the parameter dependent symmetric and uniformly positive definite matrix $\boldsymbol{A}(\omega)$ is defined similarly to a usual finite element stiffness matrix as $(\boldsymbol{A}(\omega))_{m,n} := \mathsf{a}(\omega)(\varphi_m, \varphi_n)$ with the bi-linear form $\mathsf{a}(\omega)$ given by Eq. (11). Furthermore, the right hand side (r.h.s.) is determined by $(\boldsymbol{f}(\omega))_m := \langle \ell(\omega), \varphi_m \rangle$ where the linear form $\ell(\omega)$ is given in Eq. (12), while $\boldsymbol{u}(\omega) = [u_1(\omega), \ldots, u_N(\omega)]^T$ is introduced as a vector of random coefficients as in Eq. (13).

The Eq. (14) represents a linear equation with random r.h.s. and random matrix. It is a semi-discretisation of some sort since it involves the variable ω and is still computationally intractable, as in general we need infinitely many coordinates to parametrise Ω.

2.2 Stochastic Discretisation

The semi-discretised Eq. (14) is approximated such that the stochastic input data $\boldsymbol{A}(\omega)$ and $\boldsymbol{f}(\omega)$ are described with the help of RVs of some known type. Namely, we employ a stochastic Galerkin (SG) method to do the stochastic discretisation of Eq. (14) [16, 29, 21, 2, 36, 25, 30, 3, 36, 1, 37, 32, 14, 33]. Basic convergence of such an approximation may be established via Céa's lemma [30, 27].

In order to express the unknown coefficients (RVs) $u_n(\omega)$ in Eq. (13), let us choose as the ansatz functions multivariate *Hermite* polynomials $\{H_\alpha(\boldsymbol{\theta}(\omega))\}_{\alpha \in \mathcal{J}}$ in Gaussian RVs, also known under the name *Wiener's* polynomial chaos expansion (PCE) [24, 16, 29, 30, 27]

$$u_n(\boldsymbol{\theta}) = \sum_{\alpha \in \mathcal{J}} u_n^\alpha H_\alpha(\boldsymbol{\theta}(\omega)), \qquad \text{or} \qquad \boldsymbol{u}(\boldsymbol{\theta}) = \sum_{\alpha \in \mathcal{J}} \boldsymbol{u}^\alpha H_\alpha(\boldsymbol{\theta}(\omega)), \tag{15}$$

where $\boldsymbol{u}^\alpha := [u_1^\alpha, \ldots, u_n^\alpha]^T$. The *Cameron-Martin* theorem assures us that the algebra of Gaussian variables is dense in $L_2(\Omega)$. Here the index set $\mathcal{J}$ is taken as a finite subset of $\mathbb{N}_0^{(\mathbb{N})}$, the set of all finite non-negative integer sequences, i.e. multi-indices. Although the set $\mathcal{J}$ is finite with cardinality $|\mathcal{J}| = R$ and $\mathbb{N}_0^{(\mathbb{N})}$ is countable, there is no natural order on it; and hence we do not impose one at this point.

Inserting the ansatz Eq. (15) into Eq. (14) and applying the Bubnov-Galerkin projection onto the finite dimensional subspace $\mathcal{U}_N \otimes \mathcal{S}_{\mathcal{J}}$, one requires that the weighted residuals vanish:

$$\forall \beta \in \mathcal{J}: \quad \mathbb{E}\left([\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{u}(\boldsymbol{\theta})]H_\beta(\boldsymbol{\theta})\right) = 0. \tag{16}$$

With $\boldsymbol{f}_\beta := \mathbb{E}\left(\boldsymbol{f}(\boldsymbol{\theta})H_\beta(\boldsymbol{\theta})\right)$ and $\boldsymbol{A}_{\beta,\alpha} := \mathbb{E}\left(H_\beta(\boldsymbol{\theta})\boldsymbol{A}(\boldsymbol{\theta})H_\alpha(\boldsymbol{\theta})\right)$, Eq. (16) reads:

$$\forall \beta \in \mathcal{J}: \quad \sum_{\alpha \in \mathcal{J}} \boldsymbol{A}_{\beta,\alpha}\boldsymbol{u}^\alpha = \boldsymbol{f}_\beta, \tag{17}$$

which further represents a linear, symmetric and positive definite system of equations of size $N \times R$. The system is well-posed in a sense of Hadamard since the Lax-Milgram lemma applies on the subspace $\mathcal{U}_N \otimes \mathcal{S}_{\mathcal{J}}$.

To expose the structure of and compute the terms in Eq. (17), the parametric matrix in Eq. (14) is expanded in the Karhunen-Loève expansion (KLE) [30, 28, 17, 15] as

$$\boldsymbol{A}(\boldsymbol{\theta}) = \sum_{j=0}^{\infty} \boldsymbol{A}_j \xi_j(\boldsymbol{\theta}) \tag{18}$$

with scalar RVs ξ_j. Together with Eq. (10), it is not too hard to see that $\boldsymbol{A}_j$ can be defined by the bilinear form

$$\mathsf{a}_j(v,u) := \int_{\mathcal{G}} \nabla v(x) \cdot (\kappa_j g_j(x) \nabla u(x)) \, \mathrm{d}x, \tag{19}$$

and $(\boldsymbol{A}_j)_{m,n} := \mathsf{a}_j(\varphi_m, \varphi_n)$ with $\kappa_j g_j(x)$ being the coefficient of the KL expansion of $\kappa(x,\omega)$:

$$\kappa(x,\omega) = \kappa_0(x) + \sum_{j=1}^{\infty} \kappa_j g_j(x) \xi_j(\boldsymbol{\theta}),$$

where

$$\xi_j(\boldsymbol{\theta}) = \frac{1}{\kappa_j} \left(\kappa(\cdot,\omega) - \kappa_0, g_j\right)_{L_2(\mathcal{G})} = \frac{1}{\kappa_j} \int_{\mathcal{G}} \left(\kappa(x,\omega) - \kappa_0(x)\right) g_j(x) \mathrm{d}x.$$

Now these $\boldsymbol{A}_j$ can be computed as "usual "finite element stiffness matrices with the "material properties "$\kappa_j g_j(x)$. It is worth noting that $\boldsymbol{A}_0$ is just the usual deterministic or mean stiffness matrix, obtained with the mean diffusion coefficient $\kappa_0(x)$ as parameter.
Knowing the polynomial chaos expansion of $\kappa(x,\omega) = \sum_\alpha \kappa^{(\alpha)} H_\alpha(\boldsymbol{\theta})$, compute the polynomial chaos expansion of the ξ_j as

$$\xi_j(\boldsymbol{\theta}) = \sum_{\alpha \in \mathcal{J}} \xi_j^{(\alpha)} H_\alpha(\boldsymbol{\theta}),$$

where

$$\xi_j^{(\alpha)} = \frac{1}{\kappa_j} \int_{\mathcal{G}} \kappa^{(\alpha)}(x) g_j(x) \mathrm{d}x$$

Later on we, using the PCE coefficients $\kappa^{(\alpha)}(x)$ as well as eigenfunctions $g_j(x)$, compute the following tensor approximation

$$\xi_j^{(\alpha)} \approx \sum_{l=1}^{s} (\xi_l)_j \prod_{k=1}^{\infty} (\xi_{l,\,k})_{\alpha_k},$$

where $(\xi_l)_j$ means the j-th component in the spatial space and $(\xi_{l,\,k})_{\alpha_k}$ the α_k-th component in the stochastic space.

The parametric r.h.s. in Eq. (14) has an analogous expansion to Eq. (18), which may be either derived directly from the $\mathbb{R}^N$-valued RV $\boldsymbol{f}(\omega)$—effectively a finite dimensional KLE—or from the continuous KLE of the random linear form in Eq. (12). In either case

$$\boldsymbol{f}(\omega) = \sum_{i=0}^{\infty} \sqrt{\lambda_i} \psi_i(\omega) \boldsymbol{f}_i, \tag{20}$$

where the λ_i are the eigenvalues [26, 22, 23], and, as in Eq. (18), only a finite number of terms are needed. For sparse representation of KLE see [22, 23]. The components in Eq. (17) may now be expressed as $\boldsymbol{f}_\beta = \sum_i \sqrt{\lambda_i} f_\beta^i \boldsymbol{f}_i$ with $f_\beta^i := \mathbb{E}\left(H_\beta \psi_i\right)$. Let us point out that the random variables describing the input to the problem are $\{\xi_j\}$ and $\{\psi_i\}$.

Introducing the expansion Eq. (18) into Eq. (17) we obtain:

$$\forall \beta : \quad \sum_{j=0}^{\infty} \sum_{\alpha \in \mathcal{J}} \boldsymbol{\Delta}_{\beta,\alpha}^{j} \boldsymbol{A}_j \boldsymbol{u}^\alpha = \boldsymbol{f}_\beta, \tag{21}$$

where $\boldsymbol{\Delta}_{\beta,\alpha}^{j} = \mathbb{E}\left(H_\beta \xi_j H_\alpha\right)$. Denoting the elements of the tensor product space $\mathbb{R}^N \otimes \otimes \bigotimes_{\mu=1}^M \mathbb{R}^{R_\mu}$ in an upright bold font, as for example $\mathbf{u}$, and similarly linear operators on that space, as for example $\mathbf{A}$, we may further rewrite Eq. (21) in terms of a tensor products [30, 27]:

$$\mathbf{A}\mathbf{u} := \left(\sum_{j=0}^{\infty} \boldsymbol{A}_j \otimes \boldsymbol{\Delta}^j \right) \left(\sum_{\alpha \in \mathcal{J}} \boldsymbol{u}^\alpha \otimes \boldsymbol{e}^\alpha \right) = \left(\sum_{\alpha \in \mathcal{J}} \boldsymbol{f}_\alpha \otimes \boldsymbol{e}^\alpha \right) =: \mathbf{f}, \tag{22}$$

where $\boldsymbol{e}^\alpha$ denotes the canonical basis in $\bigotimes_{\mu=1}^M \mathbb{R}^{R_\mu}$. With the help of Eq. (20) and the relations directly following it, the r.h.s. in Eq. (22) may be rewritten as

$$\mathbf{f} = \sum_{\alpha \in \mathcal{J}} \sum_{i=0}^{\infty} \sqrt{\lambda_i} f_\alpha^i \boldsymbol{f}_i \otimes \boldsymbol{e}^\alpha = \sum_{i=0}^{\infty} \sqrt{\lambda_i} \boldsymbol{f}_i \otimes \boldsymbol{g}_i, \tag{23}$$

where $\boldsymbol{g}_i := \sum_{\alpha \in \mathcal{J}} f_\alpha^i \boldsymbol{e}^\alpha$. Later on, splitting $\boldsymbol{g}_i$ further [12], obtain

$$\mathbf{f} \approx \sum_{k=1}^{R} \tilde{\boldsymbol{f}}_k \otimes \bigotimes_{\mu=1}^{M} \boldsymbol{g}_{k\mu}. \tag{24}$$

The similar splitting work, but in application in another context was done in [11, 13, 9, 10, 4]. Now the tensor product structure is exhibited also for the fully discrete counterpart to Eq. (10), and not only for the solution $\mathbf{u}$ and r.h.s. $\mathbf{f}$, but also for the operator or matrix $\mathbf{A}$.

The operator $\mathbf{A}$ in Eq. (22) inherits the properties of the operator in Eq. (10) in the sense of symmetry and positive definiteness [30, 27]. The symmetry may be verified directly from Eq. (17), while the positive definiteness follows from the Galerkin projection and the uniform convergence in Eq. (22) on the finite dimensional space $\mathbb{R}^{(N \times N)} \otimes \bigotimes_{\mu=1}^M \mathbb{R}^{(R_\mu \times R_\mu)}$. In order to make the procedure computationally feasible, of course the infinite sum in Eq. (18) has to be truncated at a finite value, say at M. The choice of M is now part of the stochastic discretisation and not an assumption.

Due to the uniform convergence alluded to above the sum can be extended far enough such that the operators $\mathbf{A}$ in Eq. (22) are uniformly positive definite with respect to the discretisation parameters [30, 27]. This is in some way analogous to the use of numerical integration in the usual FEM [34, 6, 39].

The equation 22 is solved by iterative methods in the low-rank canonical tensor format in [31]. The corresponding matlab code is implemented in [38]. Additional interesting result in [31] is the research of different strategies for the tensor-rank truncation after each iteration. Other works devoted to the research of properties of the system matrix in Eq. (25), developing of Kronecker product preconditioning and to the iterative methods to solve system in Eq. (25) are in [35, 7, 8].

Applying further splitting to $\boldsymbol{\Delta}^j$ [12], the fully discrete forward problem may finally be announced as

$$\mathbf{Au} = \left(\sum_{l=1}^{s} \tilde{\boldsymbol{A}}_l \otimes \bigotimes_{\mu=1}^{M} \boldsymbol{\Delta}_{l\mu} \right) \left(\sum_{j=1}^{r} \boldsymbol{u}_j \otimes \bigotimes_{\mu=1}^{M} \boldsymbol{u}_{j\mu} \right) = \sum_{k=1}^{R} \tilde{\boldsymbol{f}}_k \otimes \bigotimes_{\mu=1}^{M} \boldsymbol{g}_{k\mu} = \mathbf{f}, \qquad (25)$$

where $\tilde{\boldsymbol{A}}_l \in \mathbb{R}^{N\times N}$, $\boldsymbol{\Delta}_{l\mu} \in \mathbb{R}^{R_\mu \times R_\mu}$, $\boldsymbol{u}_j \in \mathbb{R}^N$, $\boldsymbol{u}_{j\mu} \in \mathbb{R}^{R_\mu}$, $\tilde{\boldsymbol{f}}_k \in \mathbb{R}^N$ and $\boldsymbol{g}_{k\mu} \in \mathbb{R}^{R_\mu}$. The similar splitting work, but in application in another context was done in [11, 13, 9, 10, 4].

3 The canonical tensor format

Let $\mathcal{T} := \bigotimes_{\mu=1}^{d} \mathbb{R}^{n_\mu}$ be the tensor space constructed from $(\mathbb{R}^{n_\mu}, \langle , \rangle_{\mathbb{R}^{n_\mu}})$ $(d \geq 3)$. From a mathematical point of view, a tensor representation U is a multilinear map from a parameter space P onto $\mathcal{T}$, i.e. $U : P \to \mathcal{T}$. The parameter space $P = \times_{\nu=1}^{D} P_\nu$ $(d \leq D)$ is the Cartesian product of tensor spaces P_ν, where in general the order of every P_ν is (much) smaller then d. Further, P_ν depends on some representation rank parameter $r_\nu \in \mathbb{N}$. A standard example of a tensor representation is the canonical tensor format.

Definition 1 (r-Terms, Tensor Rank, Canonical Tensor Format, Elementary Tensor, Representation System). *The set $\mathcal{R}_r$ of tensors which can be represented in $\mathcal{T}$ with* r-terms *is defined as*

$$\mathcal{R}_r(\mathcal{T}) := \mathcal{R}_r := \left\{ \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu} \in \mathcal{T} : v_{i\mu} \in \mathbb{R}^{n_\mu} \right\}. \qquad (26)$$

Let $v \in \mathcal{T}$. The tensor rank of v in $\mathcal{T}$ *is*

$$rank(v) := min\{ r \in \mathbb{N}_0 : v \in \mathcal{R}_r \}. \qquad (27)$$

The canonical tensor format in $\mathcal{T}$ *for variable r is defined by the mapping*

$$U_{cp} : \times_{\mu=1}^{d} \mathbb{R}^{n_\mu \times r} \to \mathcal{R}_r, \qquad (28)$$

$$\hat{v} := (v_{i\mu} : 1 \leq i \leq r,\ 1 \leq \mu \leq d) \mapsto U_{cp}(\hat{v}) := \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu}.$$

We call the sum of elementary tensors $v = \sum_{i=1}^{r} \otimes_{\mu=1}^{d} v_{i\mu} \in \mathcal{R}_r$ a tensor represented in the canonical tensor format with r terms, where an elementary tensor *is of the form $\bigotimes_{\mu=1}^{d} v_\mu \in \mathcal{R}_1$, $v_\mu \in V_\mu$. The system of vectors $(v_{i\mu} : 1 \leq i \leq r,\ 1 \leq \mu \leq d)$ is a* representation system *of v with* representation rank *r.*

Note that the representation rank refers to the representation system $(v_{i\mu} : 1 \leq i \leq r,\ 1 \leq \mu \leq d)$, not to the represented tensor. In our applications we work only with tensors represented in a tensor format. A tensor $u \in \mathcal{R}_r \subset \mathcal{T}$ with $\prod_{\mu=1}^{d} n_\mu$ entities is represented on a computer system with a representation system $\hat{u} = (u_{i\mu} \in \mathbb{R}^{n_\mu} : 1 \leq i \leq r,\ 1 \leq \mu \leq d)$ and the use of U_{cp}, i.e. $u = U_{cp}(\hat{u})$. The memory requirement for the representation system $\hat{u}$ is only $r \sum_{\mu=1}^{d} n_\mu$. Later we will see that the efficient data representation in tensor formats has several benefits for the data analysis in high dimensions. For the data analysis we need operations described in Lemma 1.

Lemma 1. *Let $r_1, r_2 \in \mathbb{N}$, $u \in \mathcal{R}_{r_1}$ and $v \in \mathcal{R}_{r_2}$. We have* *(i)* $\langle u, v\rangle_{\mathcal{T}} = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \prod_{\mu=1}^{d} \langle u_{j_1\mu}, v_{j_2\mu}\rangle_{\mathbb{R}^{n_\mu}}$*. The computational cost of* $\langle u, v\rangle_{\mathcal{T}}$ *is* $\mathcal{O}\left(r_1 r_2 \sum_{\mu=1}^{d} n_\mu\right)$*.*

(ii)$u + v \in \mathcal{R}_{r_1+r_2}$*.*

(iii)$u \odot v \in \mathcal{R}_{r_1 r_2}$*, where* $\odot$ *denotes the point wise Hadamard product. Further,* $u \odot v$ *can be computed in the canonical tensor format with* $r_1 r_2 \sum_{\mu=1}^{d} n_\mu$ *arithmetic operations.*

Proof. (i) and (ii) are trivial. For (iii), let $u = \sum_{j_1=1}^{r_1} \bigotimes_{\mu=1}^{d} u_{j_1\mu}$ and $v = \sum_{j_2=1}^{r_2} \bigotimes_{\mu=1}^{d} v_{j_2\mu}$. We have

$$u \odot v = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \left[\bigotimes_{\mu=1}^{d} u_{j_1\mu}\right] \odot \left[\bigotimes_{\mu=1}^{d} v_{j_2\mu}\right] = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \bigotimes_{\mu=1}^{d} \left[u_{j_1\mu} \odot_{n_\mu} v_{j_2\mu}\right],$$

where $\odot_{n_\mu}$ denotes the Hadamard product in $\mathbb{R}^{n_\mu}$. Obviously, we need $r_1 r_2 \sum_{\mu=1}^{d} n_\mu$ operations to determine a representation system of $u \odot v$.

Later we will use operations like the Hadamard product and the addition of tensors in the canonical format in iterative procedures. From Lemma 1 it follows that the numerical cost grows only linear respect to the order d and the representation rank of the resulting tensors will increase. The last fact makes our iterative process not feasible. Therefore, we need an approximation method which approximates a given tensor represented in the canonical format with lower rank tensors up to a given accuracy.

Definition 2 (Approximation Problem). *For given* $v \in \mathcal{R}_R$ *and* $\varepsilon > 0$ *we are looking for minimal* $r_\varepsilon \leq R$ *and* $\hat{x}^* \in \times_{\mu=1}^{d} \mathbb{R}_{\mu}^{n_\mu \times r_\varepsilon}$ *such that:*

(i) $\|v - U_{cp}(\hat{x}^*)\| \leq \varepsilon \|v\|$*,*

(ii)$\|v - U_{cp}(\hat{x}^*)\| = \text{dist}\ (v, \mathcal{R}_{r_\varepsilon}) = min_{\hat{x} \in \times_{\mu=1}^{d} \mathbb{R}^{n_\mu \times r}} \|v - U_{cp}(\hat{x})\|$*, where* $\hat{x} \in \times_{\mu=1}^{d} \mathbb{R}^{n_\mu \times r}$ *is bounded.*

The solution of this problem was already discussed in [9, 13, 11]. In the following we will denote a solution of the approximation problem from Definition 2 with $\mathfrak{App}_\varepsilon(v)$.

Note 1. Let $v \in \mathcal{R}_R$, $\varepsilon > 0$ and $U_{cp}(\hat{x}^*)$ a solution of the approximation problem as analysed in [9, 13, 11]. During the article, $U_{cp}(\hat{x}^*)$ is denote by

$$\mathfrak{App}_\varepsilon(v) := U_{cp}(\hat{x}^*). \tag{29}$$

4 Analysis of high dimensional data

In the following section let $\mathcal{I} = \times_{\mu=1}^{d} \mathcal{I}_\mu$, where $\mathcal{I}_\mu = \{i \in \mathbb{N} : 1 \leq i \leq n_\mu\}$. For the analysis of tensor structured data in high dimensions, the focus of attention is a problem depended recursively defined sequence $(u_k)_{k \in \mathbb{N}_{\geq 0}}$ represented in the canonical tensor format, i.e. we have a map $\Phi_P : \mathcal{T} \to \mathcal{T}$ such that

$$u_k := \Phi_P(u_{k-1}), \tag{30}$$

where $u_0 \in \mathcal{R}_{r_0}$ is given. The map Φ_P is constructed with the help of addition, scalar and pointwise Hadamard multiplications of tensors represented in the canonical tensor format. According to Lemma 1, the representation rank of u_k from Eq. (30) will increase. Therefore, we have to compute lower representation ranks approximations and continue the iterative process. This results in the following general inexact iteration scheme:

$$\begin{aligned} z_k &:= \Phi_P(u_{k-1}), \\ u_k &:= \mathfrak{App}_{\varepsilon_k}(z_k), \end{aligned} \tag{31}$$

Where the convergence of such inexact iterations is analysed in [20].

4.1 Computation of the maximum norm and corresponding index

We describe a approach for computing the maximum norm of $u = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu} \in \mathcal{R}_r$,

$$\|u\|_\infty := \max_{\underline{i}:=(i_1,\ldots,i_d)\in \mathrm{i}} |u_{\underline{i}}| = \max_{\underline{i}:=(i_1,\ldots,i_d)\in \mathrm{i}} \left| \sum_{j=1}^{r} \prod_{\mu=1}^{d} (u_{j\mu})_{i_\mu} \right|, \tag{32}$$

and the corresponding multi- index. Since the cardinality of $\mathcal{I}$ grows exponential with d, $\#\mathcal{I} = \prod_{\mu=1}^{d} n_\mu$, the known methods are already inefficient for small values of n_μ and d. To build an efficient algorithm we use the special tensor structure of u and show that computing $\|u\|_\infty$ is equivalent to a very simple tensor structured eigenvalue problem. Let $\underline{i}^* := (i_1^*, \ldots, i_d^*) \in \mathcal{I}$ be the index with

$$\|u\|_\infty = |u_{\underline{i}^*}| = \left| \sum_{j=1}^{r} \prod_{\mu=1}^{d} (u_{j\mu})_{i^*_\mu} \right| \text{ and } e^{(\underline{i}^*)} := \bigotimes_{\mu=1}^{d} e_{i^*_\mu},$$

where $e_{i^*_\mu} \in \mathbb{R}^{n_\mu}$ the i^*_μ-th canonical vector in $\mathbb{R}^{n_\mu}$ $(\mu \in \mathbb{N}_{\le d})$. Then for the pointwise Hadamard product of $u \odot e^{(\underline{i}^*)}$ have

$$\begin{aligned} u \odot e^{(\underline{i}^*)} &= \left[\sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu} \right] \odot \left[\bigotimes_{\mu=1}^{d} e_{i^*_\mu} \right] = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu} \odot e_{i^*_\mu} = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \left[(u_{j\mu})_{i^*_\mu} e_{i^*_\mu} \right] \\ &= \underbrace{\left[\sum_{j=1}^{r} \prod_{\mu=1}^{d} (u_{j\mu})_{i^*_\mu} \right]}_{u_{\underline{i}^*}=} \bigotimes_{\mu=1}^{d} e_{(i^*_\mu)}, \end{aligned}$$

from which follows

$$u \odot e^{(\underline{i}^*)} = u_{\underline{i}^*} e^{(\underline{i}^*)}. \tag{33}$$

Eq. (33) is an eigenvalue problem. By defining the following diagonal matrix

$$D(u) := \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \mathrm{diag}\left((u_{j\mu})_{l_\mu} \right)_{l_\mu \in \mathbb{N}_{\le n_\mu}} \tag{34}$$

with representation rank r, obtain $D(u)v = u \odot v$ for all $v \in \mathcal{T}$:

Corollary 1. *Let u, $\underline{i}^*$ and $D(u)$ are defined as described above. Then elements of u are the eigenvalues of $D(u)$ and all eigenvectors $e^{(\underline{i})}$ are of the following form:*

$$e^{(\underline{i})} = \bigotimes_{\mu=1}^{d} e_{i_\mu}, \tag{35}$$

where $\underline{i} := (i_1, \ldots, i_d) \in \mathrm{i}$ is the index of $u_{\underline{i}}$. Therefore $\|u\|_\infty$ is the largest eigenvalue of $D(u)$ with the corresponding eigenvector $e^{(\underline{i}^*)}$.

Algorithmus 1 Computing the maximum norm of $u \in \mathcal{R}_r$ by vector iteration

```
1: Choose $y_0 := \bigotimes_{\mu=1}^{d} \frac{1}{n_\mu} \underline{1}$, where $\underline{1} := (1, \dots, 1)^T \in \mathbb{R}^{n_\mu}$, $k_{\max} \in \mathbb{N}$, and take $\varepsilon := 1 \times 10^{-7}$.
2: for $k = 1, 2, \dots, k_{\max}$ do
3:
       $q_k = u \odot y_{k-1}, \quad \lambda_k = \langle y_{k-1}, q_k \rangle, \quad z_k = q_k / \sqrt{\langle q_k, q_k \rangle},$
       $y_k = \mathfrak{App}_\varepsilon(z_k).$
4: end for
```

There are different methods for the computation of the largest eigenvalue and corresponding eigenvector [18]. In this example, we simple use the power iteration to solve the eigenvalue problem. Since the tensor rank of z_k grows up monotonically, the power method described in Algorithm 1 is modified accordingly to Eq. (31). Accordingly to [19] there are

$$\mathcal{O}\left(\frac{d \log n - \log \varepsilon}{\varepsilon}\right). \tag{36}$$

iteration steps necessary to compute the maximum norm of u up to the relative error $\varepsilon \in \mathbb{R}_{>0}$. To guaranty convergence one takes the initial guess y_0 as

$$y_0 := \sum_{l_1=1}^{n_1} \cdots \sum_{l_d=1}^{n_d} \bigotimes_{\mu=1}^{d} \frac{1}{n_\mu} e_{l_\mu} = \bigotimes_{\mu=1}^{d} \left(\frac{1}{n_\mu} \sum_{l_\mu=1}^{n} e_{l_\mu} \right) = \bigotimes_{\mu=1}^{d} \frac{1}{n_\mu} \tilde{1}_\mu. \tag{37}$$

We recall that the presented method is only an approximate method to compute $\|u\|_\infty$ and $e^{(\underline{i}^*)}$. In general the vector iteration is not appropriate for solving eigenvalue problems. A possible improvement is the inverse vector iteration method, which is applied on a spectrum shift of u. Therefore is computing of the pointwise inverse necessary. Many other well-known methods require orthogonalisation, which seems for sums of elementary tensors not practicable.

4.2 Computation of the characteristic

The key object of the following approaches is a tensor which we call characteristic of $u \in \mathcal{T}$ in $I \subset \mathbb{R}$.

Definition 3 (Characteristic, Sign). *The* characteristic $\chi_I(u) \in \mathcal{T}$ *of* $u \in \mathcal{T}$ *in* $I \subset \mathbb{R}$ *is for every multi- index* $\underline{i} \in \mathcal{I}$ *pointwise defined as*

$$(\chi_I(u))_{\underline{i}} := \begin{cases} 1, \, u_{\underline{i}} \in I; \\ 0, \, u_{\underline{i}} \notin I. \end{cases} \tag{38}$$

Furthermore, the sign$(u) \in \mathcal{T}$ *is for all* $\underline{i} \in \mathcal{I}$ *pointwise defined by*

$$(\mathrm{sign}(u))_{\underline{i}} := \begin{cases} 1, & u_{\underline{i}} > 0; \\ -1, & u_{\underline{i}} < 0; \\ 0, & u_{\underline{i}} = 0. \end{cases} \tag{39}$$

Similar to the computation of the maximum norm, the computational cost of standard methods grows exponential with d, since we have to visit $\prod_{\mu=1}^{d} n_\mu$ entries of u. If u is represented in the canonical tensor format with r terms, i.e. $u = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu}$, there is a possibility to compute the characteristic $\chi_I(u)$ since there are methods to compute the sign(u).

Lemma 2. *Let $u \in \mathcal{T}$, $a, b \in \mathbb{R}$, and $\mathbb{1} = \bigotimes_{\mu=1}^{d} \tilde{1}_\mu$, where $\tilde{1}_\mu := (1, \ldots, 1)^t \in \mathbb{R}^{n_\mu}$.*

(i) If $I = \mathbb{R}_{<b}$, then we have $\chi_I(u) = \frac{1}{2}(\mathbb{1} + \operatorname{sign}(b\mathbb{1} - u))$.
(ii) If $I = \mathbb{R}_{>a}$, then we have $\chi_I(u) = \frac{1}{2}(\mathbb{1} - \operatorname{sign}(a\mathbb{1} - u))$.
(iii) If $I = (a, b)$, then we have $\chi_I(u) = \frac{1}{2}(\operatorname{sign}(b\mathbb{1} - u) - \operatorname{sign}(a\mathbb{1} - u))$.

Proof. Let $\underline{i} \in \mathcal{I}$. (i) If $u_{\underline{i}} < b \Rightarrow 0 < b - u_{\underline{i}} \Rightarrow \operatorname{sign}(b - u_{\underline{i}}) = 1 \Rightarrow \frac{1}{2}(1 + \operatorname{sign}(b - u_{\underline{i}})) = 1 = (\chi_I(u))_{\underline{i}}$. If $u_{\underline{i}} > b \Rightarrow b - u_{\underline{i}} < 0 \Rightarrow \operatorname{sign}(b - u_{\underline{i}}) = -1 \Rightarrow \frac{1}{2}(1 + \operatorname{sign}(b - u_{\underline{i}})) = 0 = (\chi_I(u))_{\underline{i}}$.
(ii) Analog to (i). (iii) Follows from (i) and (ii).

In the following part we analyse bounds for the representation rank of the characteristic $\chi_I(u)$.

Definition 4 (Cartesian Index Set, Cartesian Covering). *Let $M \subset \mathcal{I}$ be a subset of multi- indices. We call M a* Cartesian index set *if there exist $M_\mu \subset \mathcal{I}_\mu$ such that $M = \times_{\mu=1}^{d} M_\mu$. We call a set* $\operatorname{ccov}(M) = \{U \subset \mathcal{I} : U \text{ is Cartesian}\}$ *a* Cartesian covering *of M if*

$$M = \dot{\bigcup}_{U \in \operatorname{ccov}(M)} U,$$

where the symbol $\dot{\bigcup}$ stands for disjoint union.

Note that for every set $M \subseteq \mathcal{I}$ there exist a Cartesian covering.

Lemma 3. *Let $I \subseteq \mathbb{R}$, $u \in \mathcal{T}$, and $M := \operatorname{supp} \chi_I(u)$. We have*

$$rank(\chi_I(u)) \leq min\{m_1, m_2 + 1\}, \tag{40}$$

where $m_1 := min\{\#C_1 \in \mathbb{N} : C_1 \text{ is a Cartesian covering of } M\}$ and $m_2 := min\{\#C_2 \in \mathbb{N} : C_2 \text{ is a Cartesian covering of } M^c := \mathcal{I} \setminus M\}$.

Proof. Let $\{M_l = \times_{\mu=1}^{d} M_{l,\mu} : 1 \leq l \leq m_1\}$ a Cartesian covering of M and $\{N_l = \times_{\mu=1}^{d} N_{l,\mu} : 1 \leq l \leq m_2\}$ a Cartesian covering of M^c. We have

$$\begin{aligned} \chi_I(u) &= \sum_{\underline{i} \in M} \bigotimes_{\mu=1}^{d} e_{i_\mu} = \sum_{l=1}^{m_1} \sum_{i_1 \in M_{l,1}} \cdots \sum_{i_d \in M_{l,d}} \bigotimes_{\mu=1}^{d} e_{i_\mu} \\ &= \sum_{l=1}^{m_1} \bigotimes_{\mu=1}^{d} \left[\sum_{i_\mu \in M_{l,\mu}} e_{i_\mu} \right] \Rightarrow \operatorname{rank}(\chi_I(u)) \leq m_1, \end{aligned}$$

where $e_{i_\mu} \in \mathbb{R}^{n_\mu}$ is the i_μ-th canonical vector in $\mathbb{R}^{n_\mu}$. Further, we have

$$\begin{aligned} \chi_I(u) &= \mathbb{1} - \sum_{\underline{i} \in M^c} \bigotimes_{\mu=1}^{d} e_{i_\mu} = \mathbb{1} - \sum_{l=1}^{m_2} \sum_{i_1 \in N_{l,1}} \cdots \sum_{i_d \in N_{l,d}} \bigotimes_{\mu=1}^{d} e_{i_\mu} \\ &= \mathbb{1} - \sum_{l=1}^{m_2} \bigotimes_{\mu=1}^{d} \left[\sum_{i_\mu \in N_{l,\mu}} e_{i_\mu} \right] \Rightarrow \operatorname{rank}(\chi_I(u)) \leq m_2 + 1. \end{aligned}$$

The most widely used and analysed method for computing the sign function $\operatorname{sign}(A)$ of a matrix A is the Newton iteration,

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}), \quad X_0 = A. \tag{41}$$

The connection of the iteration with the sign function is not immediately obvious. The iteration can be derived by applying the Newton's method to the equation $X^2 = \boldsymbol{I}$. It is also well known that the convergence of the Newton iteration is quadratically, i.e. we have

$$\|X_{k+1} - \mathrm{sign}(A)\| \leq \frac{1}{2}\|X_k^{-1}\|\|X_k - \mathrm{sign}(A)\|^2.$$

The Newton iteration is one of the seldom circumstances in numerical analysis where the explicit computation of the inverse is required. One way to try to remove the inverse in Eq. (41) is to approximate it by one step of the Newton's method for the inverse, which has the form $Y_{k+1} = Y_k(2\boldsymbol{I} - BY_k)$ for computing B^{-1}. This leads to the Newton-Schulz iteration adapted to our tensor setting

$$u_{k+1} = \frac{1}{2}u_k \odot (3\mathbb{1} - u_k \odot u_k), \quad u_0 := u. \tag{42}$$

It is known that the Newton-Schulz iteration retains the quadratic convergence of the Newton's method. However, it is only locally convergent, with convergence guaranteed for $\|\mathbb{1} - u_0 \odot u_0\| < 1$ in some suitable norm. According to Eq. (31) the inexact Newton-Schulz iteration in tensor formats is described by Algorithm 2, where the computation of the pointwise inverse is described in Section 4.4.

Algorithmus 2 Computing $\mathrm{sign}(u)$, $u \in \mathcal{R}_r$ (Hybrid Newton-Schulz Iteration)

```
Choose $u_0 := u$ and $\varepsilon \in \mathbb{R}_+$.
while $\|\mathbb{1} - u_{k-1} \odot u_{k-1}\| < \varepsilon\|u\|$ do
    if $\|\mathbb{1} - u_{k-1} \odot u_{k-1}\| < \|u\|$ then
        $z_k := \frac{1}{2}u_{k-1} \odot (3\mathbb{1} - u_{k-1} \odot u_{k-1})$
    else
        $z_k := \frac{1}{2}(u_{k-1} + u_{k-1}^{-1})$
    end if
    $u_k := \mathfrak{App}_{\varepsilon_k}(z_k)$
end while
```

4.3 Computation of level sets, frequency, mean value, and variance

For the computation of cumulative distribution functions it is important to compute level sets of a given tensor $u \in \mathcal{T}$.

Definition 5 (Level Set, Frequency). *Let $I \subset \mathbb{R}$ and $u \in \mathcal{T}$. The* level set *$\mathcal{L}_I(u) \in \mathcal{T}$ of u respect to I is pointwise defined by*

$$(\mathcal{L}_I(u))_{\underline{i}} := \begin{cases} u_{\underline{i}}, u_{\underline{i}} \in I\ ; \\ 0, u_{\underline{i}} \notin I\ \ , \end{cases} \tag{43}$$

for all $\underline{i} \in \mathcal{I}$ The frequency *$\mathcal{F}_I(u) \in \mathbb{N}$ of u respect to I is defined as*

$$\mathcal{F}_I(u) := \#\,\mathrm{supp}\,\chi_I(u), \tag{44}$$

where $\chi_I(u)$ is the characteristic of u in I, see Definition 3.

Proposition 1. *Let $I \subset \mathbb{R}$, $u \in \mathcal{T}$, and $\chi_I(u)$ its characteristic. We have*

$$\mathcal{L}_I(u) = \chi_I(u) \odot u \tag{45}$$

and $rank(\mathcal{L}_I(u)) \leq rank(\chi_I(u))rank(u)$*. Furthermore, the frequency* $\mathcal{F}_I(u) \in \mathbb{N}$ *of* u *respect to* I *can by computed by*

$$\mathcal{F}_I(u) = \langle \chi_I(u), \mathbb{1} \rangle , \tag{46}$$

where $\mathbb{1} = \bigotimes_{\mu=1}^{d} \tilde{1}_\mu$*,* $\tilde{1}_\mu := (1, \ldots, 1)^T \in \mathbb{R}^{n_\mu}$*.*

Proposition 2. *Let* $u = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu} \in \mathcal{R}_r$*, then the mean value* $\overline{u}$ *can be computed as a scalar product*

$$\overline{u} = \left\langle \left(\sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} u_{j\mu} \right), \left(\bigotimes_{\mu=1}^{d} \frac{1}{n_\mu} \tilde{1}_\mu \right) \right\rangle = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \frac{\langle u_{j\mu}, \tilde{1}_\mu \rangle}{n_\mu} = \sum_{j=1}^{r} \prod_{\mu=1}^{d} \frac{1}{n_\mu} \left(\sum_{k=1}^{n_\mu} u_{j\mu} \right), \tag{47}$$

where $\tilde{1}_\mu := (1, \ldots, 1)^T \in \mathbb{R}^{n_\mu}$*. According to Lemma 1, the numerical cost is* $\mathcal{O}\left(r \cdot \sum_{\mu=1}^{d} n_\mu \right)$*.*

Proposition 3. *Let* $u \in \mathcal{R}_r$ *and*

$$\tilde{u} := u - \overline{u} \bigotimes_{\mu=1}^{d} \frac{1}{n_\mu} \underline{1} = \sum_{j=1}^{r+1} \bigotimes_{\mu=1}^{d} \tilde{u}_{j\mu} \in \mathcal{R}_{r+1}, \tag{48}$$

then the variance $var(u)$ *of* u *can be computed as follows*

$$\begin{aligned} var(u) &= \frac{1}{\prod_{\mu=1}^{d} n_\mu} \langle \tilde{u}, \tilde{u} \rangle = \frac{1}{\prod_{\mu=1}^{d} n_\mu} \left\langle \left(\sum_{i=1}^{r+1} \bigotimes_{\mu=1}^{d} \tilde{u}_{i\mu} \right), \left(\sum_{j=1}^{r+1} \bigotimes_{\nu=1}^{d} \tilde{u}_{j\nu} \right) \right\rangle \\ &= \sum_{i=1}^{r+1} \sum_{j=1}^{r+1} \prod_{\mu=1}^{d} \frac{1}{n_\mu} \langle \tilde{u}_{i\mu}, \tilde{u}_{j\mu} \rangle . \end{aligned}$$

According to Lemma 1, the numerical cost is $\mathcal{O}\left((r+1)^2 \cdot \sum_{\mu=1}^{d} n_\mu \right)$*.*

4.4 Computation of the pointwise inverse

Computing the pointwise inverse u^{-1} is of interest, e. g. by improved computation of the maximum norm and by iterative computations of sign(u) or $\sqrt{u}$. Let us further assume that $u_{\underline{i}} \neq 0$ for all $\underline{i} \in \mathcal{I}$. The mapping $\Phi_k : \mathcal{T} \to \mathcal{T}$ from Eq. (31) is defined as follows:

$$x \mapsto \Phi(x)_{u^{-1}} := x \odot (2\mathbb{1} - u \odot x). \tag{49}$$

This recursion is motivated through application of the Newton method on the function $f(x) := u - x^{-1}$, see [20]. After defining the error by $e_k := \mathbb{1} - u \odot x_k$, we obtain

$$e_k = \mathbb{1} - ux_k = \mathbb{1} - ux_{k-1} (\mathbb{1} + e_{k-1}) = e_{k-1} - ux_{k-1}e_{k-1} = (\mathbb{1} - ux_{k-1}) e_{k-1} = e_0^{2^k}$$

and $(x_k)_{k \in \mathbb{N}}$ converges quadratically for $\|e_0\| < 1$. Then for e_k have

$$u^{-1} - x_k = u^{-1} e_k = \left(u^{-1} - x_{k-1} \right) u \left(u^{-1} - x_{k-1} \right) = u \left(u^{-1} - x_{k-1} \right)^2 .$$

The abstract method explained in Eq. (31) is for the pointwise inverse of u specified by Algorithm 3.

Algorithmus 3 Computing u^{-1}, $u \in \mathcal{R}_r$, $u_{\underline{i}} \neq 0$ for all $\underline{i} \in \mathcal{I}$

```
1: Choose $u_0 \in \mathcal{T}$ such that $\|\mathbb{1} - u \odot u_0\| < \|u\|$ and $\varepsilon \in \mathbb{R}_+$.
2: while $\|\mathbb{1} - u \odot u_{k-1}\| < \varepsilon\|u\|$ do
3:
       $z_k := u_{k-1} \odot (2\mathbb{1} - u \odot u_{k-1}),$
       $u_k := \mathfrak{App}_{\varepsilon_k}(z_k),$
4: end while
```

Algorithmus 4 Inexact recursive iteration

```
1: Choose $u_0 \in \mathcal{T}$ and $\varepsilon \in \mathbb{R}_+$.
2: while error$(u_{k-1}) < \varepsilon$ do
3:
       $z_k := \Phi_P(u_{k-1}),$
       $u_k := \mathfrak{App}_{\varepsilon_k}(z_k),$
4: end while
```

5 Complexity Analysis

All discussed methods can be viewed as an inexact iteration procedure as mentioned in Eq. (31). For given initial guess and $\Phi_P : \mathcal{T} \to \mathcal{T}$ we have a recursive procedure defined in the following Algorithm 4. According to Lemma 1 and the problem depended definition of Φ_P the numerical cost of a function evaluation $z_k = \Phi_P(u_{k-1})$ is cheap if the tensor u_{k-1} is represented in the canonical tensor format with moderate representation rank. The dominant part of the inexact iteration method is the approximation procedure $\mathfrak{App}_{\varepsilon_k}(z_k)$.

Remark 1 *The complexity of the method* $\mathfrak{App}_{\varepsilon_k}(z_k)$ *described in [9, 11] is*

$$\mathcal{O}\left(\sum_{r=r_{k-1}}^{r_\varepsilon} m_r \cdot \left[r \cdot (r + rank(z_k)) \cdot d^2 + d \cdot r^3 + r \cdot (r + rank(z_k) + d) \cdot \sum_{\mu=1}^{d} n_\mu\right]\right) \tag{50}$$

and for the method described in [13] we have

$$\mathcal{O}\left(\sum_{r=r_{k-1}}^{r_\varepsilon} \tilde{m}_r \cdot \left[d \cdot r^3 + r \cdot (r + rank(z_k)) \cdot \sum_{\mu=1}^{d} n_\mu\right]\right) \tag{51}$$

where $r_{k-1} = rank(u_{k-1})$ *and* m_r *is the number of iterations in the regularised Newton method [9, 11] and* $\tilde{m}_r$ *is the number of iterations in the accelerated gradient method [13] for the rank-*r *approximation.*

6 Numerical Experiments

The following numerical experiments were performed on usual two-year-old PC. The multi-dimensional problem to be solved is defined in Eq. (7). The computational domain is 2D L-shape domain with $N = 557$ degrees of freedom (see Fig. 2). The number of KLE terms for q in Eq. (8) is $l_k = 10$, the stochastic dimension is $m_k = 10$ and the maximal order of Hermite polynomials is $p_k = 2$. We took the shifted lognormal distribution for $\kappa(x, \omega)$ (see Eq. (8)), i.e., $\log(\kappa(x, \omega) - 1.1)$ has

normal distribution with parameters $\{\mu = 0.5,\, \sigma^2 = 1.0\}$. The isotropic covariance function is of the Gaussian type with covariance lengths $\ell_x = \ell_y = 0.3$. The mean value and the standard deviation of $\kappa(x, \omega)$ are shown in Fig. 2.
For the right-hand side we took $l_f = 10$, $m_f = 10$ and $p_f = 2$ as well as Beta distribution with parameters $\{4, 2\}$ for random variables. The covariance function is also of the Gaussian type with covariance lengths $\ell_x = \ell_y = 0.6$. The mean value and the standard deviation of $\kappa(x, \omega)$ are shown in Fig. 3.
The Dirichlet boundary conditions in Eq. (7) were chosen as deterministic. Thus the total stochastic dimension of the solution u is $m_u = m_k + m_f = 20$, i.e. the multi- index α will consist of $m_u = 20$ indices ($\alpha = (\alpha_1, ..., \alpha_{m_u})$). The cardinality of the set of multi-indices $\mathcal{J}$ is $|\mathcal{J}| = \frac{(m_u+p_u)!}{m_u! p_u!}$, where $p_u = 2$. The solution tensor

$$u = \sum_{j=1}^{231} \bigotimes_{\mu=1}^{21} u_{j\mu} \in \mathbb{R}^{557} \otimes \bigotimes_{\mu=1}^{20} \mathbb{R}^3$$

with representation rank 231 was computed with the use of the stochastic Galerkin library [38]. The number 21 is a sum of the deterministic dimension 1 and the stochastic dimension 20. The number 557 is the number of degrees of freedom in the computational domain. In the stochastic space we used polynomials of the maximal order 2 from 20 random variables and thus the solution belongs to the tensor space $\mathbb{R}^{557} \otimes \bigotimes_{\mu=1}^{20} \mathbb{R}^3$. The mean value and the standard deviation of the solution $u(x, \omega)$ are shown in Fig. 4.

Further we computed the maximal entry $\|u\|_\infty$ of u respect to the absolute value as described in Algorithm 1. The algorithm computed after 20 iterations the maximum norm $\|u\|_\infty$ effectually. The maximal representation rank of the intermediate iterants $(u_k)_{k=1}^{20}$ was 143, where we set the approximation error $\varepsilon_k = 1.0 \times 10^{-6}$ and $(u_k)_{k=1}^{20} \subset \mathcal{R}_{143}$ is the sequence of tensors generated by Algorithm 1. Finely, we computed level sets $\operatorname{sign}(b\|u\|_\infty \mathbb{1} - u)$ for $b \in \{0.2, 0.4, 0.6, 0.8\}$. The results of the computation are documented in Table 1. The representation ranks of $\operatorname{sign}(b\|u\|_\infty \mathbb{1} - u)$ are given in the second column. In this numerical example, the ranks are smaller then 13. The iteration from Algorithm 2 determined after $k_{\max}$ steps the sign of $(b\|u\|_\infty \mathbb{1} - u)$, where the maximal representation rank of the iterants u_k from Algorithm 2 is documented in the third column. The error $\|\mathbb{1} - u_{k_{\max}} \odot u_{k_{\max}}\| / \|(b\|u\|_\infty \mathbb{1} - u)\|$ is given in the last column.

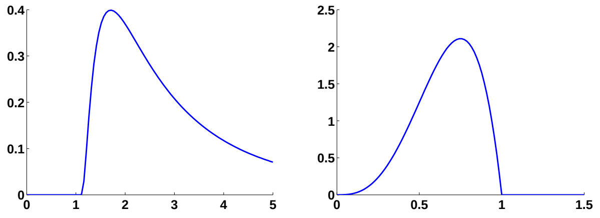

Fig. 1. Shifted lognormal distribution with parameters $\{\mu = 0.5,\, \sigma^2 = 1.0\}$ (on the left) and Beta distribution with parameters $\{4, 2\}$ (on the right).

7 Conclusion

In this work we used sums of elementary tensors for the data analysis of solutions from stochastic elliptic boundary value problems. Particularly we explained how the

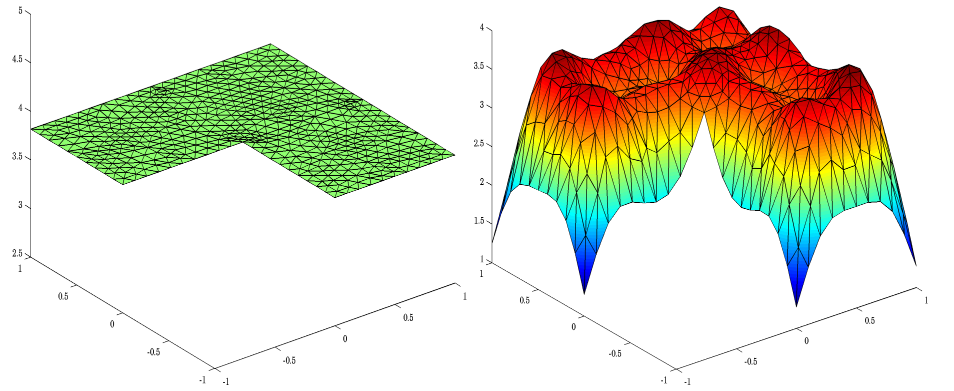

Fig. 2. Mean (on the left) and standard deviation (on the right) of $\kappa(x, \omega)$ (lognormal random field with parameters $\mu = 0.5$ and $\sigma = 1$).

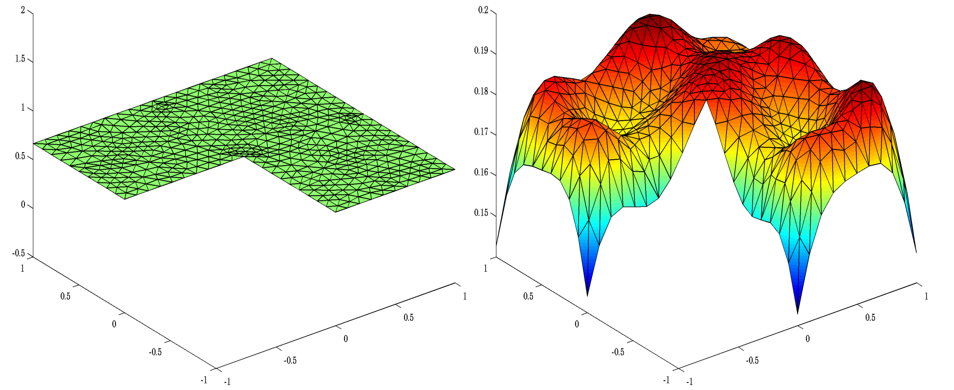

Fig. 3. Mean (on the left) and standard deviation (on the right) of $f(x, \omega)$ (beta distribution with parameters $\alpha = 4$, $\beta = 2$ and Gaussian cov. function).

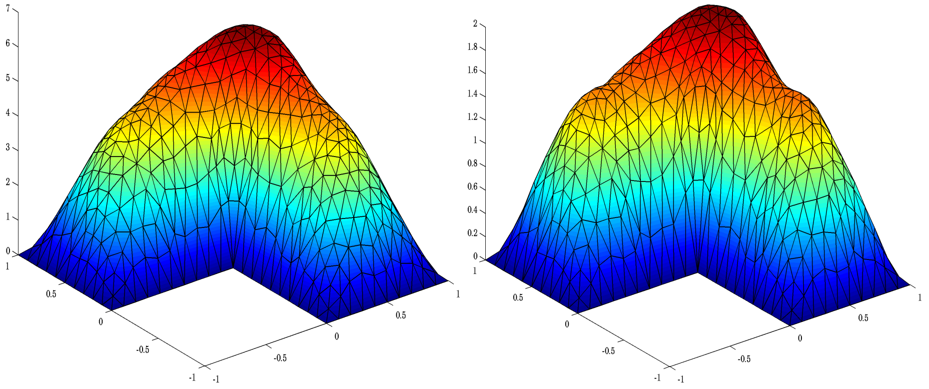

Fig. 4. Mean (on the left) and standard deviation (on the right) of the solution u.

Table 1. Computation of $\text{sign}(b\|u\|_\infty \mathbb{1} - u)$, where u is represented in the canonical tensor format with canonical rank 231, $d = 21$, $n_1 = 557$, and $p = 2$. The computing time to get any row is around 10 minutes. Note that the tensor u has $3^{20} * 557 = 1,942,138,911,357$ entries.

b	$\text{rank}(\text{sign}(b\|u\|_\infty \mathbb{1} - u))$	$\max_{1\leq k\leq k_{\max}} \text{rank}(u_k)$	$k_{\max}$	Error
0.2	12	24	12	2.9×10^{-8}
0.4	12	20	20	1.9×10^{-7}
0.6	8	16	12	1.6×10^{-7}
0.8	8	15	8	1.2×10^{-7}

new methods compute the maximum, minimum norms (Section 4.1), sign and characteristic functions (Section 4.2), level sets (Section 4.3), mean, variance (Section 4.3), and pointwise inverse (Section 4.4). In the numerical example we considered a stochastic boundary value problem in the L-shape domain with stochastic dimension 20. Table 1 illustrates computation of quantiles of the solution (via sign function). Here the computation showed that the computational ranks are of moderate size. The computing time to get any row of Table 1 is around 10 minutes. To be able to perform the offered algorithms the solution u must already be approximated in a efficient tensor format. In this article we computed the stochastic solution in a sparse data format and then approximated it in the canonical tensors format. In a upcoming paper [12] which will be submitted soon we compute the stochastic solution direct in the canonical tensor format and no transformation step is necessary.

References

1. S. Acharjee and N. Zabaras. A non-intrusive stochastic Galerkin approach for modeling uncertainty propagation in deformation processes. *Computers & Structures*, 85:244–254, 2007.
2. I. Babuška, R. Tempone, and G. E. Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J. Numer. Anal.*, 42(2):800–825, 2004.
3. I. Babuška, R. Tempone, and G. 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.
4. S. R. Chinnamsetty, M. Espig, B. N. Khoromskij, W. Hackbusch, and H. J. Flad. Tensor product approximation with optimal rank in quantum chemistry. *The Journal of chemical physics*, 127(8):084–110, 2007.
5. G. Christakos. *Random Field Models in Earth Sciences*. Academic Press, San Diego, CA, 1992.
6. P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, 1978.
7. O. G. Ernst, C. E. Powell, D. J. Silvester, and E. Ullmann. Efficient solvers for a linear stochastic Galerkin mixed formulation of diffusion problems with random data. *SIAM J. Sci. Comput.*, 31(2):1424–1447, 2008/09.
8. O. G. Ernst and E. Ullmann. Stochastic Galerkin matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1848–1872, 2010.
9. M. Espig. *Effiziente Bestapproximation mittels Summen von Elementartensoren in hohen Dimensionen*. PhD thesis, Dissertation, Universität Leipzig, 2008.
10. M. Espig, L. Grasedyck, and W. Hackbusch. Black box low tensor rank approximation using fibre-crosses. *Constructive approximation*, 2009.
11. M. Espig and W. Hackbusch. A regularized newton method for the efficient approximation of tensors represented in the canonical tensor format. *submitted Num. Math.*, 2011.
12. M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies, and P. Wähnert. Efficient approximation of the stochastic galerkin matrix in the canonical tensor format. *in preparation*.
13. M. Espig, W. Hackbusch, T. Rohwedder, and R. Schneider. Variational calculus with sums of elementary tensors of fixed rank. *paper submitted to: Numerische Mathematik*, 2009.
14. P. Frauenfelder, Ch. Schwab, and R. A. Todor. Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):205–228, 2005.
15. R. Ghanem. Ingredients for a general purpose stochastic finite element implementation. 168(1–4):19–34, 1999.
16. R. Ghanem. Stochastic finite elements for heterogeneous media with multiple random non-Gaussian properties. *Journal of Engineering Mechanics*, 125:24–40, 1999.

17. R. Ghanem and R. Kruger. Numerical solutions of spectral stochastic finite element systems. 129(3):289–303, 1996.
18. G. H. Golub and C. F. Van Loan. *Matrix Computations*. Wiley-Interscience, New York, 1984.
19. L. Grasedyck. *Theorie und Anwendungen Hierarchischer Matrizen*. Doctoral thesis, Universität Kiel, 2001.
20. W. Hackbusch, B. Khoromskij, and E. Tyrtyshnikov. Approximate iterations for structured matrices. *Numerische Mathematik*, 109:365–383, 2008. 10.1007/s00211-008-0143-0.
21. M. Jardak, C.-H. Su, and G. E. Karniadakis. Spectral polynomial chaos solutions of the stochastic advection equation. In *Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01) (Uppsala)*, volume 17, pages 319–338, 2002.
22. B. N. Khoromskij and A. Litvinenko. Data sparse computation of the Karhunen-Loève expansion. *Numerical Analysis and Applied Mathematics: Intern. Conf. on Num. Analysis and Applied Mathematics, AIP Conf. Proc.*, 1048(1):311–314, 2008.
23. B. N. Khoromskij, A. Litvinenko, and H. G. Matthies. Application of hierarchical matrices for computing Karhunen-Loève expansion. *Computing*, 84(1-2):49–67, 2009.
24. P. Krée and Ch. Soize. *Mathematics of random phenomena*, volume 32 of *Mathematics and its Applications*. D. Reidel Publishing Co., Dordrecht, 1986. Random vibrations of mechanical structures, Translated from the French by Andrei Iacob, With a preface by Paul Germain.
25. O. P. Le Maître, H. N. Najm, R. G. Ghanem, and O. M. Knio. Multi-resolution analysis of Wiener-type uncertainty propagation schemes. *J. Comput. Phys.*, 197(2):502–531, 2004.
26. H. G. Matthies. Uncertainty quantification with stochastic finite elements. 2007. Part 1. Fundamentals. Encyclopedia of Computational Mechanics, John Wiley and Sons, Ltd.
27. H. G. Matthies. Stochastic finite elements: Computational approaches to stochastic partial differential equations. *Zeitschr. Ang. Math. Mech.(ZAMM)*, 88(11):849–873, 2008.
28. H. G. Matthies, Ch. E. Brenner, Ch. G. Bucher, and C. Guedes Soares. Uncertainties in probabilistic numerical analysis of structures and solids—stochastic finite elements. 19(3):283–336, 1997.
29. H. G. Matthies and Ch. Bucher. Finite elements for stochastic media problems. *Comput. Meth. Appl. Mech. Eng.*, 168(1–4):3–17, 1999.
30. H. G. Matthies and A. Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 194(12-16):1295–1331, 2005.
31. H. G. Matthies and E. Zander. Solving stochastic systems with low-rank tensor compression. *Linear Algebra and its Applications*, In Press, Corrected Proof.
32. L. J. Roman and M. Sarkis. Stochastic Galerkin method for elliptic SPDEs: a white noise approach. *Discrete Contin. Dyn. Syst. Ser. B*, 6(4):941–955 (electronic), 2006.
33. Ch. Schwab and C. J. Gittelson. Sparse tensor discretizations of high-dimensional parametric and stochastic pdes. *Acta Numerica*, 20:291–467, 2011.
34. G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Wellesley-Cambridge Press, Wellesley, MA, 1988.
35. E. Ullmann. A Kronecker product preconditioner for stochastic Galerkin finite element discretizations. *SIAM Journal on Scientific Computing*, 32(2):923–946, 2010.
36. D. Xiu and G. E. Karniadakis. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Comput. Meth. Appl. Mech. Eng.*, 191:4927–4948, 2002.
37. X. Frank Xu. A multiscale stochastic finite element method on elliptic problems involving uncertainties. *Comput. Methods Appl. Mech. Engrg.*, 196(25-28):2723–2736, 2007.
38. E. Zander. Stochastic Galerkin library. *Technische Universität Braunschweig, http://github.com/ezander/sglib*, 2008.
39. O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*. Butterwort-Heinemann, Oxford, 5th ed., 2000.