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

Preprint ISSN 2198-5855

Non-intrusive tensor reconstruction for high dimensional random PDEs

Martin Eigel[1], Johannes Neumann[1], Reinhold Schneider[2], Sebastian Wolf[2]

submitted: October 30, 2017

[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: schneidr@math.tu-berlin.de
wolf@math.tu-berlin.de

No. 2444
Berlin 2017

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

Key words and phrases. Non-intrusive, tensor reconstruction, partial differential equations with random coefficients, tensor representation, tensor train, uncertainty quantification, low-rank.

Research of J. Neumann was funded in part by the DFG MATHEON project SE13. Research of S. Wolf was funded in part by the DFG MATHEON project SE10.

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/

Non-intrusive tensor reconstruction for high dimensional random PDEs

Martin Eigel, Johannes Neumann, Reinhold Schneider, Sebastian Wolf

ABSTRACT. This paper examines a completely non-intrusive, sample-based method for the computation of functional low-rank solutions of high dimensional parametric random PDEs which have become an area of intensive research in Uncertainty Quantification (UQ). In order to obtain a generalized polynomial chaos representation of the approximate stochastic solution, a novel black-box rank-adapted tensor reconstruction procedure is proposed. The performance of the described approach is illustrated with several numerical examples and compared to Monte Carlo sampling.

1. INTRODUCTION

For applications in engineering and the natural sciences, the inclusion of uncertainties in modeling and simulation has become a standard requirement. This can also be seen at the thriving success of the interdisciplinary field of Uncertainty Quantification (UQ), which brings together viewpoints from statistics, (numerical) analysis and other disciplines in order to accommodate and quantify random effects common in real-world settings, see e.g. [1, 2]. The treatment of randomness often is based on the description of data (e.g. coefficients, forces and domains) by means of random fields. Here, we are interested in PDE models with random data. A classical approach would then be to evaluate statistics of the solution or a quantity of interest from random draws of the stochastic data as is done with Monte Carlo sampling and modern variants such as MLMC and QMC. As an alternative, functional approximations seek to determine a surrogate model in some discretization of the function space of the problem. The most common approaches are nonintrusive interpolation methods based on sparse grids (Stochastic Collocation) and intrusive Galerkin methods (Stochastic FEM) [3–6]. A fundamental advantage of these (spectral) methods is that properties of the solution manifold, in particular sparsity and low-rank approximability, can be exploited in order to achieve optimal convergence rates possibly much higher than with sampling methods. However, one caveat is that computations can be significantly more involved. One reason for this is that a parametric representation of the stochastic fields (in countable infinite stochastic parameters) forms the basis for the functional discretization, which leads to very high dimensional systems. To alleviate this obstacle, different model reduction techniques may be employed, e.g. adaptivity and reduced basis methods. Especially promising for UQ applications are low-rank compression techniques based on hierarchical tensor representations [7–12]. It has been demonstrated that these representations can circumvent an otherwise exponential discretization complexity, allowing for the numerical treatment of high dimensional parametric PDEs which would otherwise be intractable [13–15].

In this paper, we aim to attain a functional approximation of the entire solution manifold in orthogonal polynomials (so-called generalized polynomial chaos) in parameter space and conforming finite elements (FE) in physical space. Given a tensorized basis, the functional approximation can be identified with a single high dimensional coefficient tensor. However, the exponential scaling in the order – which is equivalent to the stochastic dimensions – usually would render the accurate representation of such a tensor unfeasible. To handle this so called *curse of dimensionality*, we employ recent hierarchical tensor approximation techniques. In particular the Hierarchical Tucker [16] format and the tensor train (TT) [11] format have been shown to be well-suited for this problem class. Applications of these formats are various and can e.g. be found with high-dimensional PDEs like the Fokker Planck and the many particle Schrödinger equations, neuroscience, graph analysis, signal processing, computer vision, machine learning, and computational finance, see for instance the extensive survey of [10] and [17]. In earlier works on uncertainty quantification it was shown empirically that the operators for the parametric PDEs often exhibit a low rank structure, see for example [14, 18–20] and [12] for the analysis of the

DOI 10.20347/WIAS.PREPRINT.2444 Berlin 2017

low-rank structure of some linear operators. In [21] this setting was used to formulate a fully adaptive intrusive solution algorithm for uncertainty quantification, which provides a low rank representation of the Galerkin solution. It was empirically shown that in several cases the solution tensor of the parametric problem is efficiently representable in the low rank format, hence aleviating the curse of dimensionality, see also [22, 23] for related numerical methods and examples.

Different approaches are available to compute a tensor representation of a parametric solution. The adaptive algorithm used by [21] for a stochastic FEM requires to construct a low rank representation of the parametric operator and right hand side. Moreover, an efficient algorithm to solve the system in a low rank fashion is needed. This in particular requires adjustments for each problem setting and prevents the use of existing (possibly highly optimized) deterministic PDE solvers. The latter property is the reason why these methods are called intrusive. As a remedy to the mentioned difficulties, in this work, we propose a completely non-intrusive black box reconstruction algorithm, which attempts to recover the complete solution tensor from a set of Monte Carlo measurements of the solution. These can be obtained without further modifications from any existing PDE solver of choice. Additionally, since we do not require a specific choice of measurements, this method can also be applied for already available data, without the need for recomputations. The presented approach is based on a so-called tensor reconstruction, which is an generalization of the well-known matrix reconstruction [24–26]. We refer the interested reader to [22, 27] for related approaches.

This paper starts with a brief description of the considered UQ setting for parametric random PDEs in Section 2. In particular, the representation of the stochastic solution in high dimensional tensor spaces is given. Section 3 provides an overview of the Tensor Train (TT) format. The rank adaptive alternating steepest descent algorithm employed in the reconstruction algorithm is described in Section 4. Its performance is illustrated in the numerical experiments in Section 5. Finally Section 6 summarizes the results.

2. Uncertainty Quantification with high dimensional Tensors

In this section, we briefly review the setting of PDEs with countable (infinite) stochastic parameters as commonly considered in UQ, cf. [2, 28] and also [6, 21, 29, 30] for results in the adaptive Galerkin setting. We are concerned with the computation of a feasible (regarding the complexity) representation of the parametric solution $u(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{V} = \mathcal{X} \otimes \mathcal{Y}$ of an abstract problem

$$\mathcal{D}(u;\boldsymbol{y}) = 0.$$

Here, $\mathcal{D}$ encodes the model in a physical domain $D \subset \mathbb{R}^d$, $d = 1,2,3$, and $\boldsymbol{y}$ is a M (possible infinite $M = \infty$) dimensional parameter vector describing the data. We assume a product parameter domain Γ and the components of $\boldsymbol{y}$ to correspond to i.i.d. random variables with associated joint product density γ. For the PDE models we consider, a classical Karhunen-Loève representation of some finite variance stochastic field $a(\boldsymbol{x},\boldsymbol{y})$ with $\boldsymbol{x} \in D$ and $\boldsymbol{y} \in \Gamma$ is assumed, see e.g. [2, 31, 32]. In particular, for the numerical experiments, we examine the two prototypical cases of affine and lognormal dependence on $\boldsymbol{y}$ such that either

$$a(\boldsymbol{x},\boldsymbol{y}) = a_0 + \sum_{m=1}^{M} a_m(\boldsymbol{x}) y_m, \tag{1}$$

with $\Gamma = [-1,1]^M$ and $y_i \sim \mathcal{U}(-1,1)$, $i = 1,\dots,M$, or

$$a(\boldsymbol{x},\boldsymbol{y}) = \exp\left(\sum_{m=1}^{M} a_m(\boldsymbol{x}) y_m\right),$$

with $\Gamma = \mathbb{R}^M$ and $y_m \sim \mathcal{N}(0,1)$, $m = 1,\dots,M$. We are interested in the solution $u \in L^2(\Gamma,\gamma;\mathcal{X}) \simeq \mathcal{X}\otimes\mathcal{Y}$, usually with $\mathcal{X} = H_0^1(D)$ and $\mathcal{Y} = L^2(\Gamma,\gamma)$. An approximation u_N in a discrete subspace $\mathcal{V}_N \subset \mathcal{V}$ is given by

$$u \approx u_N(\boldsymbol{x},\boldsymbol{y}) = \sum_{\boldsymbol{\alpha}\in\mathcal{I}_N} u_{\boldsymbol{\alpha}}(\boldsymbol{x}) P_{\boldsymbol{\alpha}}(\boldsymbol{y}).$$

Here, $\mathcal{I}_N \subset \mathcal{F}$ is some finite subset of the set of multiindices with finite support $\mathcal{F}$. For $\boldsymbol{\alpha}\in\mathcal{I}_N$, the orthonormal tensorized basis polynomials $P_{\boldsymbol{\alpha}}(\boldsymbol{y}) = \prod_{m=1}^M P_{\alpha_m}(y_m)$ are based on univariate Legendre or Hermite polynomials (depending on the distribution of the y_m) P_k of degree k with $\mathbb{E}_\gamma[P_k P_l] = \delta_{k,l}$, $k,l\in\mathbb{N}$. The coefficients $u_{\boldsymbol{\alpha}}$ are functions in a discrete conforming FE space $\mathcal{X}_p(\mathcal{T}) = \operatorname{span}\{\varphi_j\}_{j=1,\dots,S} \subset \mathcal{X}$ of order p on a triangulation $\mathcal{T}$ of the domain D. Hence, $\mathcal{V}_N = \operatorname{span}\{\varphi_j \otimes P_{\boldsymbol{\alpha}} \mid j = 1,\dots,S, \boldsymbol{\alpha}\in\mathcal{I}_N\}$ with FE basis functions $\varphi_i \in \mathcal{X}_p(\mathcal{T})$.

Since Γ exhibits a Cartesian product structure, the underlying energy space $\mathcal{V}$ is a (countable) tensor product of Hilbert spaces endowed with a cross-norm. A truncated expansion of the field $a(\boldsymbol{x},\boldsymbol{y})$, i.e. $M<\infty$, yields a finite tensor product which makes it amenable for computations in tensor formats. This leads to the discrete representation

$$u_N(\boldsymbol{x},\boldsymbol{y}) = \sum_{j=1}^{S}\sum_{\boldsymbol{\alpha}\in\mathcal{I}_N} \mathcal{W}(j,\boldsymbol{\alpha})\varphi_j(\boldsymbol{x})P_{\boldsymbol{\alpha}}(\boldsymbol{y}), \tag{2}$$

with coefficient tensor $\mathcal{W}\in\mathbb{R}^{S\times q_1\times\cdots\times q_M}$ where q_m-1 is the maximal polynomial degree in the m-th mode, encoded in $\mathcal{I}_N$. The non-intrusive black box computation of $\mathcal{W}$ is the subject of the following sections.

3. Hierarchical Tensor Decompositions

As shown in the previous section the complete (parametric) solution to a given differential equation can be represented in one high dimensional tensor $\mathcal{W}\in\mathbb{R}^{N_\mathcal{X}\times q_1\times\cdots\times q_M}$. The main drawback of this representation is the exponential scaling of higher order tensors, often referred to as the *curse of dimensionality*. This means that even for a small local dimensions $n_1,\dots,n_d$ with $n := \max(n_i)$ and moderate order d the storage complexity of $\mathcal{O}(n^d)$ quickly becomes unfeasible. This problem is not specific to the uncertainty quantification application but an intrinsic property of high dimensional tensor spaces. In this section we present one possible remedy to this major complexity problem, namely the Hierarchical Tucker (HT) format and in particular its special case the Tensor Train (TT) format. Both formats can be considered as high dimensional generalizations of the popular matrix singular value decomposition (SVD) and allow for very efficient storage and computations on a rich set of *low rank* tensors.

3.1. Notation.

We first provide a brief introduction of the tensor notation used in this work. For an in-depth discussion of the following concepts and tensor spaces in general, we refer to the monograph of Hackbusch [33]. For the sake of simplicity tensors in this work are simply regarded as d-dimensional real arrays in $\mathbb{R}^{n_1\times\dots\times n_d}$. The individual dimensions of the array are refereed to as the modes of the tensor. It is often expedient to combine certain subsets of modes into single modes and thereby interpret a tensor as a lower order object, usually a vector or a matrix. This operation is formalized as the matrifications of a tensor.

Definition 1 (Matrification)**.** Given a tensor space $\mathbb{R}^{n_1\times\dots\times n_d}$, let $\alpha\subseteq\{1,\dots,d\}$ denote a subset of the modes and let $\beta = \{1,\dots,d\}\setminus\alpha$ be its complement. Define $n_\alpha := \prod_{j\in\alpha} n_j$ and $n_\beta := \prod_{j\in\beta} n_j$.

Given a bijection

$$\begin{aligned}\varphi : \mathbb{N}_{n_1} \times \ldots \times \mathbb{N}_{n_d} &\to \mathbb{N}_{n_\alpha} \times \mathbb{N}_{n_\beta} \\ (i_1, \ldots, i_d) &\mapsto (\varphi_\alpha(i_k | k \in \alpha), \varphi_\beta(i_k | k \in \beta)),\end{aligned}$$

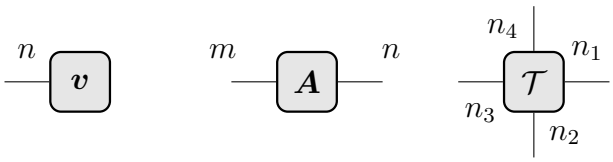
the mapping

$$\begin{aligned}\mathrm{Mat}_\alpha : \mathbb{R}^{n_1 \times \ldots \times n_d} &\to \mathbb{R}^{n_\alpha \times n_\beta} \\ \mathcal{T} &\mapsto \mathrm{Mat}_\alpha(\mathcal{T}) \\ \mathrm{Mat}_\alpha(\mathcal{T})[i,j] &:= \mathcal{T}[\varphi^{-1}(i,j)] \qquad\qquad \forall i \in \mathbb{N}_{n_\alpha}, j \in \mathbb{N}_{n_\beta}\end{aligned}$$

is called an α-matrification.

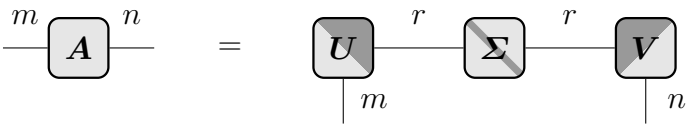
In this work the choice of the bijection φ does not matter since it is supposed to be the same in all instances. We henceforth assume a simple lexicographical ordering of the indices.

To denote the contraction of two tensors along certain modes we use the $\circ$ symbol indexed by the nodes that are contracted. For instance, $\mathcal{A} \circ_{(i,j),(k,l)} \mathcal{B}$ means that the i-th and k-th mode of $\mathcal{A}$ and $\mathcal{B}$ are contracted, as well as the j-th and l-th mode. If no indices are given, a contraction of the last mode of the left operand and the first mode of the right operand is assumed. As writing this for larger tensor expressions quickly becomes cumbersome we also use a diagrammatic notation to visualize these contractions. In this notation a tensor is depicted as a block with edges corresponding to each of its modes. If appropriate, the local dimension of the corresponding mode is given as well. From left to right the following diagrams show this for an order one tensor (vector) $\boldsymbol{v} \in \mathbb{R}^n$, an order two tensor (matrix) $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and an order four tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$.



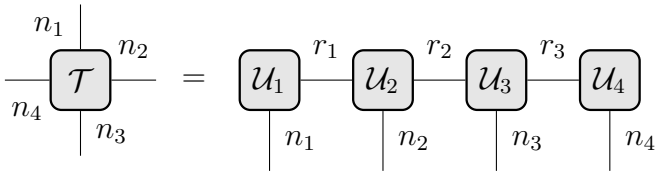
If a contraction is performed between two modes the corresponding edges are joined. The following shows this exemplarily for the inner product of two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ and a matrix-vector product with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{v} \in \mathbb{R}^n$.

There are two special cases concerning orthogonal and diagonal matrices. If a specific matrification of a tensor yields an orthogonal or diagonal matrix, the tensor is depicted by a half filled square (orthogonal) or a circle with a diagonal bar (diagonal). The half filling and the diagonal bar both divide the square in two halves. The edges joined to either half correspond to the mode sets of the matrification, which yields the orthogonal or diagonal matrix. As an example, the diagrammatic notation can be used to depict the singular value decomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with rank r, as shown in the following.



3.2. **The Tensor Train Format.** In this section we introduce the *Tensor Train* format. The formulation used in this work was first introduced by Oseledets [11]. However, an equivalent formulation was known in quantum physic for quite some time, see e.g. [34] for an overview. The TT format is closely related to the more general hierarchical Tucker (HT) format introduced by Hackbusch and Kühn [16] and is often considered to be a special case thereof. See [35] for the details of their relation.

The fundamental idea of the TT format is to separate the modes of a higher order tensor into d tensors of small order two or three. This results in a tensor network, that is exemplarily shown for an order four tensor $\mathcal{T} = \mathcal{U}_1 \circ \mathcal{U}_2 \circ \mathcal{U}_3 \circ \mathcal{U}_4$ in the following diagram.



Formally the tensor train format can be defined as follows.

Definition 2 (Tensor Train Format)**.** Let $\mathcal{T} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ be a tensor of order d. A factorization

$$\mathcal{T} = \mathcal{U}_1 \circ \mathcal{U}_2 \circ \ldots \circ \mathcal{U}_{d-1} \circ \mathcal{U}_d$$

into component tensors $\mathcal{U}_1 \in \mathbb{R}^{n_1 \times r_1}$, $\mathcal{U}_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}, i = 2, \ldots, d-1$ and $\mathcal{U}_d \in \mathbb{R}^{r_{d-1}, n_d}$, is called a tensor train (TT) representation of $\mathcal{T}$. The tuple of the dimensions $(r_1, \ldots, r_{d-1})$ of the component tensors is called the representation rank and is associated with the specific representation. The *tensor train rank* (TT-rank) is defined as the minimal rank tuple $\boldsymbol{r} = (r_1, \ldots, r_{d-1})$ such that there exists a TT representation of $\mathcal{T}$ with representation rank equal to $\boldsymbol{r}$.

Theorem 3 (Oseledets [11])**.** *Every tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ admits a TT-representation with minimal representation rank $\boldsymbol{r} = (r_1, \ldots, r_{d-1})$. Furthermore this minimal rank $\boldsymbol{r}$, which thereby is the TT-rank of $\mathcal{T}$, is related to the ranks of a certain matrification of $\mathcal{T}$ via*

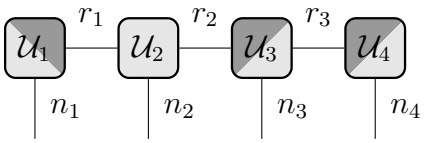
$$r_i = \operatorname{rank}\left(\operatorname{Mat}_{(1,2,\ldots,i)}(\mathcal{T})\right) \ .$$

Storing a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ with TT-rank $\boldsymbol{r} = (r_1, \ldots, r_{d-1})$ in a minimal TT-representation requires $\Theta(dnr^2)$ memory for the component tensors, where $n = \max_i(n_i)$ and $r = \max_i(r_i)$. This means the storage requirement scales only linearly in the order, which is a drastical reduction compared to the exponential scaling for general tensors. A similar scaling can also be obtained for many common operations on tensors in the TT representation, like sums, entriewise products and certain contractions, which makes the TT format a very powerful tool when dealing with high dimensional data.

It can easily be shown that a the TT representation is not unique, e.g. for any invertible matrix $\boldsymbol{A} \in \mathbb{R}^{r_1 \times r_i}$ there holds

$$\begin{aligned}\mathcal{T} &= \mathcal{U}_1 \circ \mathcal{U}_2 \circ \ldots \circ \mathcal{U}_{d-1} \circ \mathcal{U}_d \\ &= \mathcal{U}_1 \circ \mathcal{U}_2 \circ \ldots \circ \mathcal{U}_i \circ \boldsymbol{A} \circ \boldsymbol{A}^{-1} \circ \mathcal{U}_{i+1} \circ \ldots \circ \mathcal{U}_{d-1} \circ \mathcal{U}_d \\ &= \mathcal{U}_1 \circ \mathcal{U}_2 \circ \ldots \circ \tilde{\mathcal{U}}_i \circ \tilde{\mathcal{U}}_{i+1} \circ \ldots \circ \mathcal{U}_{d-1} \circ \mathcal{U}_d\end{aligned}$$

It is often convenient to ”normalize“ the representation by enforcing that for a given $1 \leq i \leq d$ all component tensors with smaller index are left orthogonal, i.e. the matrification which combines all but the last mode is an orthogonal matrix, and all component tensors with larger index are right orthogonal, i.e. the matrification which combines all but the first mode is an orthogonal matrix. The component tensor which is not orthogonal is called the core – or more precisely is said to carry the core. The following diagram shows a normalized order four tensor, where the second component carries the core.



In the following we assume that all TT representations are normalized in this way, unless explicitly stated otherwise. Note however that even with this normalization the representation is still not unique, as for example choosing $\boldsymbol{A}$ to be an orthogonal matrix is still valid in the above example.

As a natural generalization of the low rank matrix manifold [36], it is shown by Holtz et al. [37] that the set $\mathcal{M}_{\boldsymbol{r}}\left(\mathbb{R}^{n_1\times\ldots\times n_d}\right)$ of tensors with TT rank $\boldsymbol{r}$ forms a manifold.

Theorem 4 (Tensor Train Manifold [37])**.** *For fixed dimensions $n_1,\ldots,n_d$ and a fixed rank $\boldsymbol{r} = (r_1,\ldots,r_{d-1})$ the set $\mathcal{M}_{\boldsymbol{r}}\left(\mathbb{R}^{n_1\times\ldots\times n_d}\right)$ is a smooth manifold of dimension*

$$\dim\left(T_{\boldsymbol{r}}\right) = \sum_{i=1}^{d} r_{i-1}n_i r_i - \sum_{i=1}^{d-1} r_i^2\,,$$

where by convention $r_0 = r_d = 1$.

This manifold structure of $\mathcal{M}_{\boldsymbol{r}}\left(\mathbb{R}^{n_1\times\ldots\times n_d}\right)$ will be important later for the derivation of our reconstruction algorithm. For this we also need the following result on the structure of the tangent space of $\mathcal{M}_{\boldsymbol{r}}\left(\mathbb{R}^{n_1\times\ldots\times n_d}\right)$.

Theorem 5 (Tangent Space [37])**.** *The tangent space $\mathbb{T}_{\mathcal{X}}\mathcal{M}_{\boldsymbol{r}}\left(\mathbb{R}^{n_1\times\ldots\times n_d}\right)$ at a point $\mathcal{X} = \mathcal{U}_1\circ\ldots\circ\mathcal{U}_d \in \mathcal{M}_{\boldsymbol{r}}\left(\mathbb{R}^{n_1\times\ldots\times n_d}\right)$ is given by*

$$\mathbb{T}_{\mathcal{X}} = V_1\oplus\ldots\oplus V_d$$

where, for $j = 1,\ldots,d-1$,

$$\begin{aligned} V_j = \{&\mathcal{U}_1\circ\ldots\circ\mathcal{U}_{j-1}\circ\Delta\circ\mathcal{U}_{j+1}\circ\ldots\circ\mathcal{U}_d \\ &\mid \Delta\in\mathbb{R}^{r_{j-1}\times n_i\times r_j}, \mathcal{U}_i\circ_{(1,2),(1,2)}\Delta = \mathbf{0}\big\}\,, \\ V_d = \big\{&\mathcal{U}_1\circ\ldots\circ\mathcal{U}_{d-1}\circ\Delta \mid \Delta\in\mathbb{R}^{r_{d-1}\times n_d\times 1}\big\}\,. \end{aligned}$$

Here it is assumed that the decomposition is normalized such that $\mathcal{U}_1,\ldots,\mathcal{U}_{d-1}$ are left orthogonal.

In this theorem the construction of the spaces V_j ensures that all spaces are orthogonal to each other. For this work this orthogonality is not required and in the following we will instead use the spaces

$$V_j = \left\{\mathcal{U}_1\circ\ldots\circ\mathcal{U}_{j-1}\circ\Delta\circ\mathcal{U}_{j+1}\circ\ldots\circ\mathcal{U}_d \middle|\, \Delta\in\mathbb{R}^{r_{j-1}\times n_i\times r_j}\right\}\,,$$

for all $1\leq j\leq d$. It is easy to verify that $\mathbb{T}_{\mathcal{X}} = V_1+\ldots+V_d$ still holds, but the spaces are no longer complementary. The decomposition of the tangent space into these subspaces has the advantage that explicit projectors on these spaces can be given as shown by Lubich et al. [38].

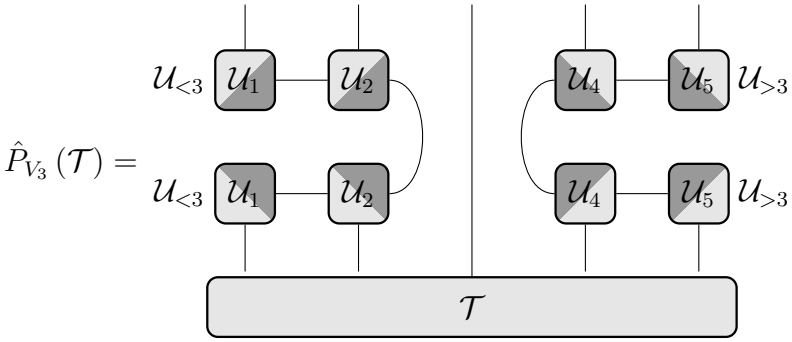
Theorem 6 (Tangent Subspace Projection [38])**.** *The projectors $\hat{P}_{V_j}$ onto the the subspaces V_i of the tangent space $\mathbb{T}_{\mathcal{X}}\mathcal{M}_{\boldsymbol{r}}\left(\mathbb{R}^{n_1\times\ldots\times n_d}\right)$ at a point $\mathcal{X}\in\mathcal{M}_{\boldsymbol{r}}\left(\mathbb{R}^{n_1\times\ldots\times n_d}\right)$ can be explicitly given. For this, define the tensors*

$$\begin{aligned} \mathcal{U}_{<j} &= \mathcal{U}_1\circ\mathcal{U}_2\circ\ldots\circ\mathcal{U}_{j-1}, \\ \mathcal{U}_{>j} &= \mathcal{U}_{j+1}\circ\mathcal{U}_{j+2}\circ\ldots\circ\mathcal{U}_d. \end{aligned}$$

Here, the $\mathcal{U}_k$ are the respectively left or right orthogonal components of a TT representation of $\mathcal{X}$ such that the core is at position j. The projector is then given by

$$\begin{aligned}\hat{P}_{V_j}(\mathcal{T}) = \left(\mathcal{U}_{<j} \circ_{(j),(j)} \mathcal{U}_{<j}\right) &\circ_{(1,\dots,j-1),(1,\dots,j-1)} \mathcal{T} \\ &\circ_{(j+1,\dots,d),(1,\dots,d-j+1)} \left(\mathcal{U}_{>j} \circ_{(1),(1)} \mathcal{U}_{>j}\right).\end{aligned}$$

The composition of the projector can be expressed intuitively in the graphical notation, shown exemplarily for an order five tensor in the following



These subspace projectors conclude the theoretical tools needed for the reconstruction approach described in the next section.

4. Reconstruction Approach

The aspiration of our reconstruction based approach is in the setting of section 2, to recover the complete solution tensor $\mathcal{W} \in \mathbb{R}^{N_{\mathcal{X}} \times q_1 \times \cdots \times q_M}$, using only a set of Monte Carlo measurements. If the solution is a general tensor this would require at least as many measurements as the tensor has degrees of freedom, i.e. the exponential number of its entries. In almost all cases this would be unfeasible. However it was empirically shown e.g. in [21–23] that for stochastic PDEs the resulting solution tensor can exhibit a very low TT-rank, see also the references given in the introduction. Using a low rank assumption, the degrees of freedom of the solution reduce to the dimension of the corresponding TT manifold. In particular, this number scales only linearly in the order. Following the ideas of compressive sensing, one could therefore hope to reconstruct the solution from a rather small number of measurements. Formally, one could try to reconstruct the solution by solving the following minimization problem,

$$\begin{aligned}&\text{minimize TT-Rank}\left(\mathcal{X}\right) \\ &\text{subject to } \hat{\mathcal{A}}\left(\mathcal{X}\right) = \boldsymbol{b}\,.\end{aligned} \tag{3}$$

where $\hat{\mathcal{A}}$ is a linear operator corresponding to the parameters of the measurements and $\boldsymbol{b}$ corresponds to the measured values, i.e. the solution for these parameters. In our particular setting, the samples are indexed by k and $\hat{\mathcal{A}}$ and $\boldsymbol{b}$ are given as

$$\begin{aligned}\left(\hat{\mathcal{A}}(\mathcal{X})\right)_{j,k} &= \left(\mathcal{X} \circ_M \left(\boldsymbol{\xi}_1^{(k)} \otimes \ldots \otimes \boldsymbol{\xi}_M^{(k)}\right)\right)[j] \\ \boldsymbol{b}_{j,k} &= \sum_{\boldsymbol{\alpha} \in \mathcal{I}_N} \mathcal{W}(j, \boldsymbol{\alpha}) P_{\boldsymbol{\alpha}}(\boldsymbol{y})\,,\end{aligned}$$

for $k = 1, \dots, N$ and $j = 1, \dots, S$. Here, assuming a function basis $\{P_j\}$ as described in Section 2, $\boldsymbol{\xi}_m^{(k)}$ are vectors of base evaluations for the k-th sample (i.e. a parameter realization),

$$\boldsymbol{\xi}_m^{(k)} := \begin{pmatrix} P_1(y_m^{(k)}) \\ P_2(y_m^{(k)}) \\ \vdots \\ P_{q_i}(y_m^{(k)}) \end{pmatrix} \in \mathbb{R}^{q_m}.$$

Problem (3) is exactly the so called low rank tensor recovery setting, which is the high dimensional generalization of the well established matrix reconstruction setting. In fact, for tensors of order two, this is the matrix reconstruction problem. In this case it is proven that under some assumptions on the measurement operator $\mathcal{A}$ there is a unique solution to this problem, which is attained by solving the convex trace norm minimization problem. The important result is that even for random measurements, the number of needed measurments scales only as $(m + n)r$, see e.g. [39] and [24]. Unfortunately, equivalent results could not yet be obtained for higher order tensors and finding a convex relaxation of the rank minimization problem proves to be a highly non-trivial task, see [40]. Nevertheless, there is much numerical evidence that also for higher order tensors, reconstruction from incomplete measurements is possible, see e.g. [26, 41]. Similar to the referenced works we "bypass" the rank minimization by starting at a rank one tensor and employ a heuristical rank adaption during the optimization. Let us initially assume that the optimal rank $\boldsymbol{r}*$ is known, then (3) can be recast as an optimization problem on the low rank manifold $\mathcal{M}_{\boldsymbol{r}*}$,

$$\begin{aligned} &\text{minimize } J(\mathcal{X}) = \|\mathcal{A}(\mathcal{X}) - \boldsymbol{b}\| \\ &\text{subject to } \mathcal{X} \in \mathcal{M}_{\boldsymbol{r}*} \,. \end{aligned}$$

There are several algorithms which solve such optimization problems posed on the low rank manifold, see e.g. [17] for a recent survey. Most popular is probably the alternating least squares (ALS) algorithm [42], which optimizes one of of the components $\mathcal{U}_i$ of the TT representation at a time. A second more recent approach are Riemannian optimization techniques which directly exploit the manifold structure of $\mathcal{M}_{\boldsymbol{r}*}$, see for example [41] for an application to tensor completion. For this work we use an alternating steepest decent algorithm which can be considered to lie in between the two groups of algorithms.

4.1. **Alternating Steepest Descent.** The alternating steepest descent algorithm used in this work can be seen as a generalization of the alternating directional fitting (ADF) algorithm developed by Grasedyck et al. [43] for tensor completion. Our derivation however is quite different and may provide some additional insight into the algorithm.

As established in Theorem 4, the set of rank $\boldsymbol{r}$ tensors $\mathcal{M}_{\boldsymbol{r}}$ forms a smooth embedded manifold. For the optimization algorithm we will in particular exploit the orthogonal decomposition of the tangent space (Theorem 5) and the closed form for the projector $\hat{P}_{\mathbb{T}_{\mathcal{X}}\mathcal{M}_{\boldsymbol{r}}}$ of Theorem 6. The essential idea is to create an iteration by projecting the gradient onto *one* of the spaces V_j at a time and perform an update in this direction. The main advantage is that independent of the step size such an update never leaves the manifold [1], because only the core component is altered in each step. In contrast to Riemannian optimization algorithms which rely on a retraction to stay on the manifold, this allows a direct calculation of the optimal step size with respect to the global optimization problem 1. Given the

[1] That is, the rank cannot increase. Theoretically the update could end in a critical point, i.e. a tensor with smaller rank.

Algorithm 1: Abstract Alternating Steepest Descent

Input: Inital guess $\mathcal{X}_0$, operator $\hat{\mathcal{A}}$ and right hand side $\boldsymbol{b}$
for $i = 1, 2, \dots$ *until converged* **do**
  for $j = 1, \dots, d$ **do**
    Calculate the projected gradient $\hat{P}_j(\nabla J(\mathcal{X}_i))$
    Calculate the optimal step size $\alpha_i = \frac{\left\langle \hat{\mathcal{A}}(\mathcal{X}_i) - \boldsymbol{b},\, \hat{\mathcal{A}}(\hat{P}_j(\nabla J(\mathcal{X}_i)))\right\rangle}{\left\| \hat{\mathcal{A}}(\hat{P}_j(\nabla J(\mathcal{X}_i)))\right\|_2^2}$
    Update $\mathcal{X}_{i+1} = \mathcal{X}_i - \alpha_i \hat{P}_j(\nabla J(\mathcal{X}_i))$

current iterate $\mathcal{X}_i$ and an update direction $\mathcal{Y}_i$ the iteration step size is given as

$$\alpha_i = \frac{\left\langle \hat{\mathcal{A}}(\mathcal{X}_i) - \boldsymbol{b},\, \hat{\mathcal{A}}(\hat{P}_j(\mathcal{Y}))\right\rangle}{\left\| \hat{\mathcal{A}}(\hat{P}_j(\mathcal{Y}))\right\|_2^2} .$$

Here $\hat{P}_j$ are orthogonal projectors onto the spaces V_j from Theorem 6. An abstract outline of the complete alternating steepest decent algorithm is provided in Listing 1.

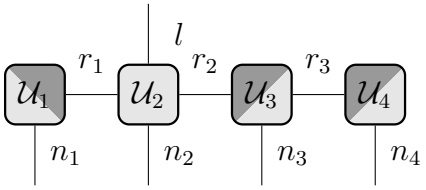
4.2. **Block Alternating Steepest Descent.** The alternating steepest descent, as well as Riemannian optimization algorithms on the TT-manifold, only works with a fixed TT-rank. In order to tackle the tensor reconstruction problem from the beginning of this section, one either has to guess the optimal rank a priori or use a rank adaptation strategy. The first approach is naturally very efficient if a sufficiently accurate rank estimate is available. However for most settings, including the UQ setting considered here, such an estimate is not available. The simplest rank adaptation strategy is to increase the TT-ranks by adding a rank increasing random perturbation to the current iterate, whenever the manifold optimization gets stuck in a (local) minimum. More elaborated but also more expensive approaches estimate the potential gain of a rank increase at each position and increase the rank by adding a low rank truncation of the real gradient at this position, see also [44].

Instead we propose a new rank adaptation strategy for the tensor recovery setting which is an adoption from the block-ALS algorithms, introduced by Dolgov et al. [45] for the calculation of extreme eigenvalues. This strategy allows for the selective increase of ranks at beneficial positions and also provides a specific subspace for the increase. The essential idea is to split the measurements into two (or possibly more) sets, and use the block-TT format to optimize separate iterates using one set of measurements each. This block-TT was introduced by Dolgov et al. [45] and allows to represent several tensors in the TT-format which share the same orthogonal basis for all but one mode.

Definition 7 (Block Tensor Train [45])**.** The tensors $\mathcal{T}_1, \dots, \mathcal{T}_l \in \mathbb{R}^{n_1 \times \dots \times n_d}$ are said to be in the block-TT format with core position μ if there exist left orthogonal $\mathcal{U}_1, \dots U_{\mu-1}$, right orthogonal $\mathcal{U}_{\mu+1}, \dots, \mathcal{U}_d$ and $\mathcal{U}_{\mu,1}, \dots, \mathcal{U}_{\mu,l}$ such that for all p

$$\mathcal{T}_k = \mathcal{U}_1 \circ \dots \mathcal{U}_{\mu-1} \circ \mathcal{U}_{\mu,p} \circ \mathcal{U}_{\mu+1} \circ \dots \circ \mathcal{U}_d .$$

That is $\mathcal{T}_1, \dots, \mathcal{T}_l$ can be represented in the same TT-representation with only the core tensor changing. It will be convenient to use a core tensor $\mathcal{U}_\mu \in \mathbb{R}^{r_{\mu-1} \times n_\mu \times l \times r_\mu}$ of order four, where the third mode encodes the extra index. In the following this is exemplary shown in the diagrammatic notation.



In the proposed block ASD, this format is used to perform two separate ASD iterations in parallel, while enforcing a coherent basis for all components which are not currently optimized. To this end the set of measurements is randomly split into two equally large subsets. This gives rise to two independent operators $\hat{\mathcal{A}}_p$ and right hand sides $\boldsymbol{b}_p$ and thereby to two independent optimization problems

$$\begin{aligned} &\text{minimize } J_p(\mathcal{X}) := \|\mathcal{A}_p(\mathcal{X}) - \boldsymbol{b}_p\| \\ &\text{subject to } \mathcal{X} \in \mathcal{M}_{\boldsymbol{r}} \,. \end{aligned}$$

For each of these optimization problems the ASD is used, with the addition that when moving the core after each micro iteration, the block TT-format is preserved, i.e. a common orthogonal basis is calculated. This is achieved naturally by moving the core including the extra ”block“ mode by calculating a complete singular value decomposition. The heuristic is that if the basis of the true solution is found, all iterates stay within this subspace and moving the core will not increase the rank. However if the current basis is too small, the different sub-problems evolve in different directions, which can be quantified by the singular value decomposition and is used subsequently to increase the rank when moving the core. As this is done separately for each core position, this allows a selective rank increase and also the prescription of an accuracy for which an increase or decrease should be carried out. It has shown to be beneficial to slowly increase this accuracy whenever the algorithms gets stuck with the current rank. The complete algorithm is given in Listing 2. For both sets, the algorithms starts with the average of all measurements as an inital guess. Additionally to the two sets of measurements used for the optimization, a third set containing a small amount of the measurements is used to prevent over-fitting when increasing the rank.

The most important property of the block alternating steepest descent algorithm is that it can be implemented very efficiently for the problem at hand. Surprisingly the rank adaptation method via the block-TT format be added without any significant increase of the computational complexity compared to the standard ASD. In particular, by storing and reusing partial applications of the projectors and measurement operators $\mathcal{A}_p$, the computational complexity of one iteration scales as $\mathcal{O}(NSr^2 + NMqr^2)$, where N is the number of samples, $r = \max_i(r_i)$ and $q = \max_i(q_i)$.

4.3. **TT Postprocessing.** Many statistical properties of the solution are directly available from the TT reconstruction. Assuming orthognormal Hermite or Legendre polynomials, depending on the random distributions, the expectation value is directly given as $\mathbb{E}[u(\boldsymbol{x})] = \sum_{j=1}^{S} \mathcal{W}[j, 0, \dots, 0]\varphi_j(\boldsymbol{x})$, which is obtained from the TT representation with a negligible cost. Another benefit of the tensor representation is that the solution for any given parameter set, i.e. a realization of the random variables, can be calculated at significantly reduced costs by contracting $\mathcal{W}$ with the corresponding parameter vectors. The computational complexity of these contractions scales as $\mathcal{O}(Sr + Mr^2n)$. For reasonable ranks this is a drastic cost reduction compared to actually solving the PDE with the corresponding parameters as done in Monte Carlo sampling.

Algorithm 2: Block Alternating Steepest Descent for Recovery

Input: Measured parameters $\left(\boldsymbol{\xi}_1^{(k)}, \ldots, \boldsymbol{\xi}_M^{(k)}\right)_{k=1,\ldots,N}$ and values $\boldsymbol{b}$.

Set $\varepsilon = 0.1$

Set $\boldsymbol{U}[j,p] := \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{b}_{j,k}, \quad p = 1,2$
Set $\mathcal{X} := \mathcal{U}_1 \circ \ldots \circ \mathcal{U}_{M+1} = \boldsymbol{U} \otimes \boldsymbol{e}_1 \otimes \ldots \otimes \boldsymbol{e}_1$
With $\mathcal{X}_p$ denote $\mathcal{X}$ with the block mode fixed at index p

Random partition measurements, yielding $\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2, \hat{\mathcal{A}}_{test}, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_{test}$

for $i = 0, 1, 2, \ldots$ *until stopping criterion* **do**
  for $\mu = 1, \ldots, M+1$ **do**
    for $p = 1, 2$ **do**
      Calculate $\mathcal{D} = \mathcal{U}_{<j} \circ_{(1,\ldots\mu-1),(1,\ldots,\mu-1)} \nabla J_p(\mathcal{X}_p) \circ_{(\mu+1,\ldots,M+1),(2,\ldots,M+1-\mu)} \mathcal{U}_{>j}$
      Calculate $\alpha = \frac{\left\langle \hat{\mathcal{A}}_p(\mathcal{X}_p) - \boldsymbol{b}_p,\ \hat{\mathcal{A}}_p(\mathcal{U}_0 \circ \ldots \circ \mathcal{U}_{\mu-1} \circ \mathcal{D} \circ \mathcal{U}_{\mu+1} \circ \ldots \circ \mathcal{U}_M) \right\rangle}{\left\| \hat{\mathcal{A}}_p(\mathcal{U}_0 \circ \ldots \circ \mathcal{U}_{\mu-1} \circ \mathcal{D} \circ \mathcal{U}_{\mu+1} \circ \ldots \circ \mathcal{U}_M) \right\|_2^2}$
      Update $\mathcal{U}_\mu[:,:,p,:] \mathrel{-}= \alpha \mathcal{D}[:,:,:]$
    if $\mu < M$ **then**
      Calculate SVD with tolerance ε: $\mathcal{U} \circ \boldsymbol{\Sigma} \circ \mathcal{V} = \mathrm{Mat}_{(1,2)}(\mathcal{U}_\mu)$
      Set $\mathcal{U}_\mu = \mathcal{U}$
      Set $\mathcal{U}_{\mu+1} = \boldsymbol{S} \circ \mathcal{V} \circ \mathcal{U}_{\mu+1}$
      Shuffle the block mode to the third position in $\mathcal{U}_{\mu+1}$
  for $\mu = M, M-1, \ldots, 0$ **do**
    for $p = 1, 2$ **do**
      Calculate $\mathcal{D} = \mathcal{U}_{<j} \circ_{(1,\ldots\mu-1),(1,\ldots,\mu-1)} \nabla J_p(\mathcal{X}_p) \circ_{(\mu+1,\ldots,M+1),(2,\ldots,M+1-\mu)} \mathcal{U}_{>j}$
      Calculate $\alpha = \frac{\left\langle \hat{\mathcal{A}}_p(\mathcal{X}_p) - \boldsymbol{b}_p,\ \hat{\mathcal{A}}_p(\mathcal{U}_0 \circ \ldots \circ \mathcal{U}_{\mu-1} \circ \mathcal{D} \circ \mathcal{U}_{\mu+1} \circ \ldots \circ \mathcal{U}_M) \right\rangle}{\left\| \hat{\mathcal{A}}_p(\mathcal{U}_0 \circ \ldots \circ \mathcal{U}_{\mu-1} \circ \mathcal{D} \circ \mathcal{U}_{\mu+1} \circ \ldots \circ \mathcal{U}_M) \right\|_2^2}$
      Update $\mathcal{U}_\mu[:,:,p,:] \mathrel{-}= \alpha \mathcal{D}[:,:,:]$
    if $\mu > 0$ **then**
      Calculate SVD with tolerance ε: $\mathcal{U} \circ \boldsymbol{\Sigma} \circ \mathcal{V} = \mathrm{Mat}_{(1,3)}(\mathcal{U}_\mu)$
      Set $\mathcal{U}_\mu = \mathcal{U}$
      Set $\mathcal{U}_{\mu-1} = \mathcal{U}_{\mu+1} \circ \mathcal{V} \circ \boldsymbol{S}$
      Shuffle the block mode to the third position in $\mathcal{U}_{\mu+1}$
  if $J((\mathcal{X}_1 + \mathcal{X}_2)/2)$ *did not decrease for 10 iterations* **then**
    Set $\varepsilon = \varepsilon/2$
  if $J_{test}((\mathcal{X}_1 + \mathcal{X}_2)/2)$ *did not decrease for 100 iterations* **then**
    Stop the iteration
return $(\mathcal{X}_1 + \mathcal{X}_2)/2$

5. Numerical Experiments

This section is concerned with illustrating the performance of the proposed tensor reconstruction approach for a set of parametric benchmark problems. To assess the quality of the functional tensor representation of the stochastic solution, the error of the expectation and the variance with respect to a reference Monte Carlo reference solution is depicted. In all settings the reference solution is

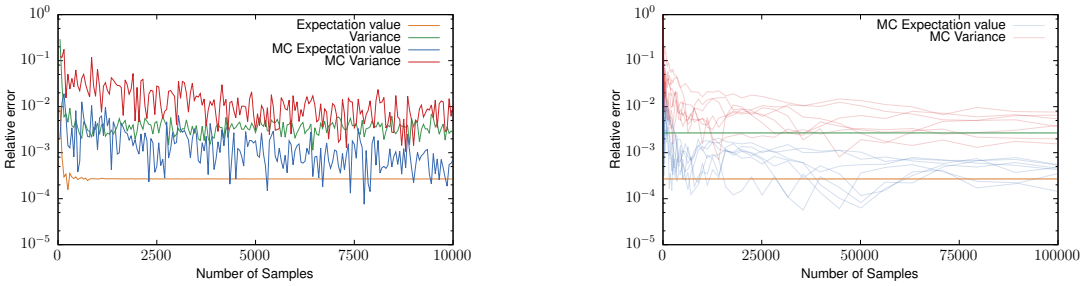


Figure 1. Comparison of the reconstruction results for the parametric PDE setting (I) a MC simulation using the exact same samples (left) and with several MC simulations using various numbers of samples (right).

calculated from $100,000$ samples.[2] In all numerical experiments, the domain is $D = (0,1)^2$ and a lowest-order conforming Finite Element discretization is employed. The sought stochastic solutions admit the representation (2) in a tensor basis with either Legendre or Hermite polynomials as described in Section 2. For the reconstruction a maximal degree of 4 (Hermite) and 12 (Legendre) is allowed for each polynomial basis. The maximal allowed rank is set to be 40, which for most problems is not a restriction. Additionally we prevent excessive rank increases by allowing each rank to increase only once every 10 iterations. The examined problems comprise linear and nonlinear PDEs with data expansions based on uniform and Gaussian random variables in Section 5.1. Moreover, results for the so-called cookie problem are reported on in Section 5.2. Since the numerical results exhibit similar behavior, we postpone the discussion to the concluding Section 6.

The computation of the FE solutions are performed with the freely available `FEniCS` software packages. For the tensor reconstructions the `xerus` C++ library [46] is used, which is freely available and also contains the complete implementation of our block ASD algorithm.

5.1. **Linear and Nonlinear Parametric PDEs.** As in [6, 21, 29, 30], we assume a random field expansion (1) given by[3]

$$a_0 \equiv 1, \quad a_m(x) := \alpha_m \cos(2\pi\beta_1(m)x_1)\cos(2\pi\beta_2(m)x_2), \quad m \in \mathbb{N},$$

where $\alpha_m = \frac{9}{10\zeta(2)} m^{-2}$ with the Riemann zeta function ζ. Moreover,

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

with $k(m) := \lfloor -1/2 + \sqrt{1/4 + 2m} \rfloor$. We consider the following stationary parametric PDEs with homogeneous Dirichlet boundary conditions and $a(x,y)$ truncated after $M = 10$ terms in all cases:

(I) **Diffusion** (affine)
$$-\nabla \cdot (a(x,y)\nabla u(x,y)) = 1$$

(II) **Diffusion** (lognormal)
$$-\nabla \cdot (\exp(a(x,y))\nabla u(x,y)) = 1$$

(III) **Nonlinear Diffusion** (affine)
$$-\nabla \cdot \left(\left(\frac{a(x,y)}{10} + u(x,y) \right)^2 \nabla u(x,y) \right) = 1$$

[2] We aim at significantly improving the quality of the reference solution using an increased number of samples for the revision of this paper.

[3] This choice corresponds to the slow decay experiments e.g. in [6, 29].

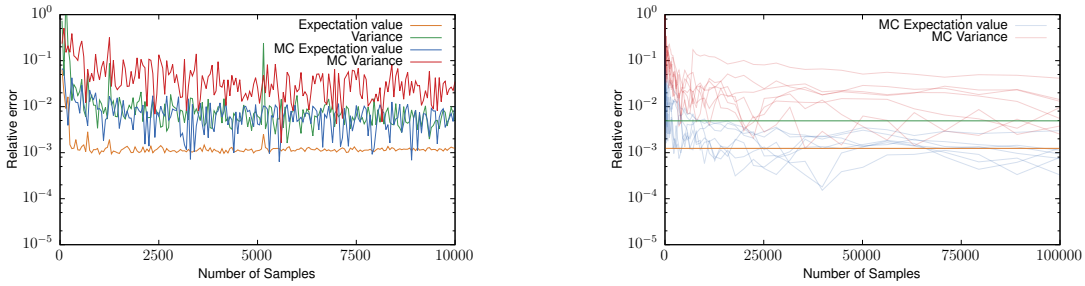


Figure 2. Comparison of the reconstruction results for the parametric PDE setting (II) a MC simulation using the exact same samples (left) and with several MC simulations using various numbers of samples (right).

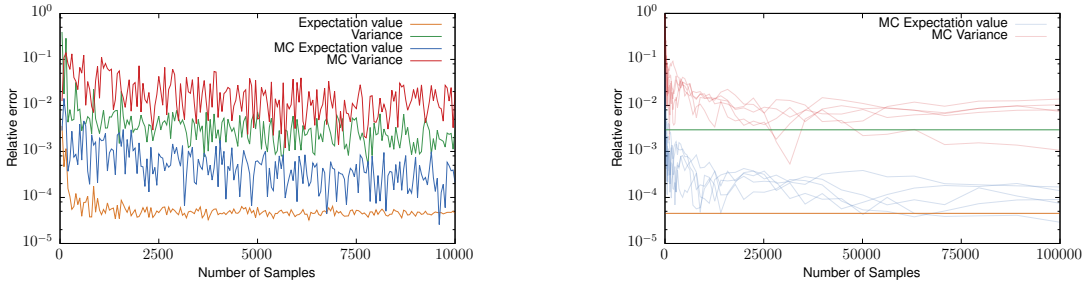


Figure 3. Comparison of the reconstruction results for the parametric PDE setting (III) with a MC simulation using the exact same samples (left) and with several MC simulations using various numbers of samples (right).

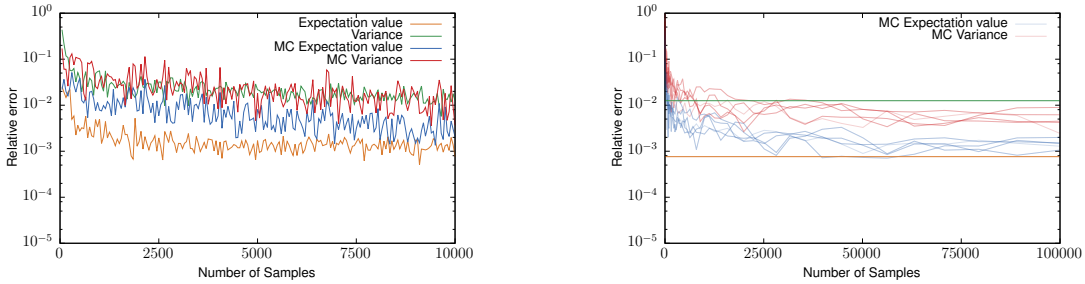


Figure 4. Comparison of the reconstruction results for the parametric PDE setting (IV) with a MC simulation using the exact same samples (left) and with several MC simulations using various numbers of samples (right).

(IV) **Convection-Diffusion** (affine, SUPG stabilized FEM)

$$-\nabla \cdot (\kappa \nabla u(x,y)) + \beta \cdot \nabla u(x,y) = 1,$$

with $\kappa = 10^{-2}$ and $\beta = \begin{pmatrix} 0.99 - a(x,y) & 0.99 - |a(x,y)| \end{pmatrix}^\mathsf{T}$.

In figures 1 to 4 the relative errors for the expectation value and variance obtained by the reconstruction is compared to Monte Carlo simulations.

5.2. **A Cookie Problem.** Let 4 subdomains of D be given by discs D_k ($k = 1, \dots, 4$) with fixed radius $r = 3/16$ and centers $c = \begin{pmatrix} i/4 & j/4 \end{pmatrix}^\mathsf{T}$ for $i, j \in \{1, 3\}$. The considered problem depends on $y = \begin{pmatrix} y_1 & y_2 & y_3 & y_4 \end{pmatrix}^\mathsf{T}$ with $y_k \sim \mathcal{U}(1/2, 3/2)$, has homogeneous Dirichlet boundary conditions

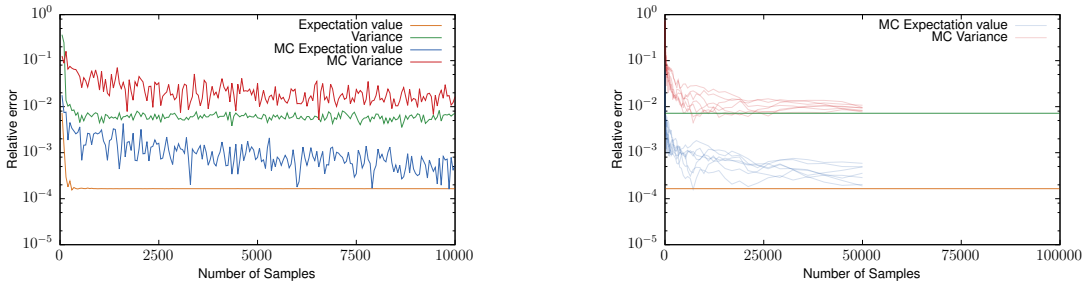


Figure 5. Comparison of the reconstruction results for the cookie setting with a MC simulation using the exact same samples (left) and with several MC simulations using various numbers of samples (right).

and is given by

$$-\nabla \cdot (\kappa(x,y)\nabla u(x,y)) = 1,$$

where $\kappa|_{D\setminus\cup_{k=1,\dots,4}D_k} = 1$ and $\kappa|_{D_k} = y_k$.

The relative errors for the expectation value and variance obtained by the reconstruction is compared to Monte Carlo simulations in figure 5.

6. Conclusions

This work is concerned with the derivation of a completely non-intrusive tensor recovery approach for the generation of numerical parametric PDE solutions in tensor Hilbert spaces. Some comments about the proposed method and its numerical performance as demonstrated in the last section are in order:

- The approach can be conceived as an alternative to a Monte Carlo method, based on principally the same (but possibly much fewer) samples of realizations of the random parametric solution. In contrast to many other approaches no specific choice of samples is required at all.
- The reconstruction only requires the solution samples and information on the distribution. The tensor rank is adaptively determined within the reconstruction algorithm. Therefore this reconstruction approach can be used as an effective black-box replacement for Monte Carlo simulations.
- For the examined linear and nonlinear parametric PDEs in Section 5 with affine and non-affine coefficient fields, it is apparent (and mostly known from earlier references) that the solution manifolds have low rank. The same holds true for the cookie problem.
- An observation common to the examined examples is that the tensor reconstruction requires orders of magnitude fewer samples when compared to a classical Monte Carlo solution. This is shows for the convergence of the expectation of the solution and its variance.[4]

References

[1] M. D. Gunzburger, C. G. Webster, and G. Zhang.
„Stochastic finite element methods for partial differential equations with random input data“.
In: *Acta Numer.* 23 (2014), pp. 521–650.

[4]Note that the MC reference solutions obtained from 2×10^5 samples is not sufficiently accurate to allow for a correct convergence plot of the significantly more accurate TT solution. We intend to use a more accurate reference solution in the revision of this paper.

[2] C. Schwab and C. J. Gittelson.
„Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs“.
In: *Acta Numer.* 20 (2011), pp. 291–467.
[3] I. Babuška, F. Nobile, and R. Tempone.
„A stochastic collocation method for elliptic partial differential equations with random input data“.
In: *SIAM J. Numer. Anal.* 45.3 (2007), pp. 1005–1034.
[4] F. Nobile, R. Tempone, and C. G. Webster. „A sparse grid stochastic collocation method for partial differential equations with random input data“.
In: *SIAM J. Numer. Anal.* 46.5 (2008), pp. 2309–2345.
[5] I. Babuška, R. Tempone, and G. E. Zouraris. „Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation“.
In: *Comput. Methods Appl. Mech. Engrg.* 194.12-16 (2005), pp. 1251–1294.
[6] M. Eigel, C. J. Gittelson, C. Schwab, and E. Zander. „Adaptive stochastic Galerkin FEM“.
In: *Comput. Methods Appl. Mech. Engrg.* 270 (2014), pp. 247–269.
[7] W. Hackbusch. „Numerical tensor calculus“. In: *Acta Numer.* 23 (2014), pp. 651–742.
[8] W. Hackbusch. *Tensor spaces and numerical tensor calculus*. Vol. 42.
Springer Series in Computational Mathematics. Springer, Heidelberg, 2012, pp. xxiv+500.
[9] W. Hackbusch and R. Schneider. „Tensor spaces and hierarchical tensor representations“.
In: *Extraction of Quantifiable Information from Complex Systems*. Springer, 2014, pp. 237–261.
[10] L. Grasedyck, D. Kressner, and C. Tobler.
„A literature survey of low-rank tensor approximation techniques“.
In: *GAMM-Mitteilungen* 36.1 (2013), pp. 53–78.
[11] I. V. Oseledets. „Tensor-train decomposition“.
In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317.
[12] M. Bachmayr, A. Cohen, and W. Dahmen.
„Parametric PDEs: Sparse or low-rank approximations?“
In: *arXiv preprint arXiv:1607.04444* (2016).
[13] M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies, and P. Wähnert.
„Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats“.
In: *Comput. Math. Appl.* 67.4 (2014), pp. 818–829.
[14] M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies, and P. Wähnert.
Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats. Preprint.
Max Planck Institute for Mathematics in the Sciences, 2012.
[15] S. Dolgov, B. N. Khoromskij, A. Litvinenko, and H. G. Matthies.
„Computation of the response surface in the tensor train data format“. Submitted / accepted.
2014.
[16] W. Hackbusch and S. Kühn. „A New Scheme for the Tensor Representation“. English.
In: *Journal of Fourier Analysis and Applications* 15.5 (2009), pp. 706–722.
[17] M. Bachmayr, R. Schneider, and A. Uschmajew. „Tensor Networks and Hierarchical Tensors for the Solution of High-Dimensional Partial Differential Equations“.
In: *Foundations of Computational Mathematics* (2016), pp. 1–50.
[18] B. N. Khoromskij and C. Schwab.
„Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs“.
In: *SIAM Journal on Scientific Computing* 33.1 (2011), pp. 364–385.
[19] H. G. Matthies, A. Litvinenko, O. Pajonk, B. V. Rosić, and E. Zander.
„Parametric and uncertainty computations with tensor product representations“.
In: *Uncertainty Quantification in Scientific Computing*. Springer, 2012, pp. 139–150.
[20] B. N. Khoromskij and I. V. Oseledets.
„Quantics-TT collocation approximation of parameter-dependent and stochastic elliptic PDEs“.
In: *Comput. Methods Appl. Math.* 10.4 (2010), pp. 376–394.

[21] M. Eigel, M. Pfeffer, and R. Schneider.
„Adaptive stochastic Galerkin FEM with hierarchical tensor representations“.
In: *Numerische Mathematik* (2015), pp. 1–39.

[22] S. Dolgov, B. N. Khoromskij, A. Litvinenko, and H. G. Matthies.
„Polynomial Chaos Expansion of random coefficients and the solution of stochastic partial differential equations in the Tensor Train format“. In: *arXiv preprint arXiv:1503.03210* (2015).

[23] S. Dolgov and R. Scheichl.
„A hybrid Alternating Least Squares–TT Cross algorithm for parametric PDEs“.
In: *arXiv preprint arXiv:1707.04562* (2017).

[24] B. Recht, M. Fazel, and P. A. Parrilo.
„Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization“.
In: *SIAM review* 52.3 (2010), pp. 471–501.

[25] E. J. Candès and T. Tao. „The power of convex relaxation: Near-optimal matrix completion“.
In: *IEEE Transactions on Information Theory* 56.5 (2010), pp. 2053–2080.

[26] H. Rauhut, R. Schneider, and Z. Stojanac.
„Low rank tensor recovery via iterative hard thresholding“.
In: *arXiv preprint arXiv:1602.05217* (2016).

[27] M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies, and E. Zander.
„Efficient analysis of high dimensional data in tensor formats“. In: *Sparse Grids and Applications*. Springer, 2013, pp. 31–56.

[28] G. J. Lord, C. E. Powell, and T. Shardlow. *An Introduction to Computational Stochastic PDEs*. Cambridge Texts in Applied Mathematics. Cambridge University Press, New York, 2014, pp. xii+503.

[29] M. Eigel, C. J. Gittelson, C. Schwab, and E. Zander. „A convergent adaptive stochastic Galerkin finite element method with quasi-optimal spatial meshes“.
In: *ESAIM: Mathematical Modelling and Numerical Analysis* 49.5 (2015), pp. 1367–1398.

[30] M. Eigel and C. Merdon. „Local equilibration error estimators for guaranteed error control in adaptive stochastic higher-order Galerkin FEM“. In: *WIAS Preprint* 1997 (2014).

[31] M. Loève. *Probability Theory. I*. Fourth. Graduate Texts in Mathematics, Vol. 45. Springer-Verlag, New York-Heidelberg, 1977, pp. xvii+425.

[32] C. Schwab and R. A. Todor.
„Karhunen-Loève approximation of random fields by generalized fast multipole methods“.
In: *J. Comput. Phys.* 217.1 (2006), pp. 100–122.

[33] W. Hackbusch. *Tensor spaces and numerical tensor calculus*. Vol. 42. Springer Science & Business Media, 2012.

[34] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. „Matrix product state representations“.
In: *arXiv preprint quant-ph/0608197* (2006). arXiv: quant-ph/0608197 [quant-ph].

[35] L. Grasedyck and W. Hackbusch.
„An introduction to hierarchical (H-) rank and TT-rank of tensors with examples“.
In: *Computational Methods in Applied Mathematics Comput. Methods Appl. Math.* 11.3 (2011), pp. 291–304.

[36] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[37] S. Holtz, T. Rohwedder, and R. Schneider. „On manifolds of tensors of fixed TT-rank“.
In: *Numerische Mathematik* 120.4 (2012), pp. 701–731.

[38] C. Lubich, I. V. Oseledets, and B. Vandereycken. „Time integration of tensor trains“.
In: *SIAM Journal on Numerical Analysis* 53.2 (2015), pp. 917–941.

[39] E. J. Candes and Y. Plan. „Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements“.
In: *IEEE Transactions on Information Theory* 57.4 (2011), pp. 2342–2359.

[40] S. Oymak, A. Jalali, M. Fazel, Y. C. Eldar, and B. Hassibi.
„Simultaneously structured models with application to sparse and low-rank matrices“.
In: *IEEE Transactions on Information Theory* 61.5 (2015), pp. 2886–2908.
[41] D. Kressner, M. Steinlechner, and B. Vandereycken.
„Low-rank tensor completion by Riemannian optimization“.
In: *BIT Numerical Mathematics* 54.2 (2014), pp. 447–468.
[42] S. Holtz, T. Rohwedder, and R. Schneider.
„The alternating linear scheme for tensor optimization in the tensor train format“.
In: *SIAM Journal on Scientific Computing* 34.2 (2012), A683–A713.
[43] L. Grasedyck, M. Kluge, and S. Krämer.
„Alternating directions fitting (ADF) of hierarchical low rank tensors“. In: *Preprint* 149 (2013).
[44] L. Grasedyck and S. Krämer.
„Stable ALS Approximation in the TT-Format for Rank-Adaptive Tensor Completion“.
In: *arXiv preprint arXiv:1701.08045* (2017).
[45] S. V. Dolgov, B. N. Khoromskij, I. V. Oseledets, and D. V. Savostyanov.
„Computation of extreme eigenvalues in higher dimensions using block tensor train format“.
In: *Computer Physics Communications* 185.4 (2014), pp. 1207–1216.
[46] B. Huber and S. Wolf. *Xerus - A General Purpose Tensor Library*. 2014–2017.