Hierarchical Tensor Approximation of Output Quantities of Parameter-Dependent PDEs

Jonas Ballani[*] **and** **Lars Grasedyck**[†]

Bericht Nr. 385 **März 2014**

Key words: hierarchical Tucker, low rank tensor, SPDE,
cross approximation, sampling

AMS Subject Classifications: 65C20, 65C50, 65D15

Institut für Geometrie und Praktische Mathematik
RWTH Aachen

Templergraben 55, D–52056 Aachen (Germany)

[*] MATHICSE-ANCHP, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. Email:jonas.ballani@epfl.ch. Financial support from the DFG SPP-1324 under grant GR 3179/2-2 gratefully acknowledged.

[†] Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, 52056 Aachen, Germany. Email: lgr@igpm.rwth-aachen.de

Hierarchical tensor approximation of output quantities of parameter-dependent PDEs

Jonas Ballani [*], Lars Grasedyck [†]

March 14, 2014

Parametric PDEs appear in a large number of applications, as, e.g., in uncertainty quantification or optimisation. In many cases, one is interested in scalar output quantities induced by the parameter-dependent solution. The output can be interpreted as a tensor living on a high-dimensional parameter space. Our aim is to adaptively construct an approximation of this tensor in a data-sparse hierarchical tensor format. Once this approximation from an offline computation is available, the evaluation of the output for any parameter value becomes a cheap online task. Moreover, the explicit tensor representation can be used to compute stochastic properties of the output in a straightforward way. The potential of this approach is illustrated by numerical examples.

1 Introduction

Partial differential equations depending on parameters are encountered in many areas of practical interest, as, e.g., in uncertainty quantification or optimisation. In an abstract form, a parametric PDE can be expressed by

$$\mathcal{D}(u,p) = f \tag{1}$$

in a bounded domain $\Omega \subset \mathbb{R}^\ell$ $(\ell = 1,2,3)$, where $\mathcal{D}$ is a partial differential operator that depends on d parameters represented by $p = (p_1,\dots,p_d) \in P \subset \mathbb{R}^d$. More generally, the right-hand side f, the domain Ω, and suitable boundary conditions may also depend on p. Assuming the well-posedness of problem (1), each instance of $p \in P$ defines a unique solution $u = u(p)$.

[*] MATHICSE-ANCHP, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. Email: jonas.ballani@epfl.ch. Financial support from the DFG SPP-1324 under grant GR 3179/2-2 gratefully acknowledged.

[†] Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, 52056 Aachen, Germany. Email: lgr@igpm.rwth-aachen.de

In many cases, one is not directly interested in the solution u itself but rather in scalar output quantities $\phi : P \to \mathbb{R}$ defined by

$$\phi(p) = \Phi(u(p)), \quad p \in P, \tag{2}$$

where Φ is a (not necessarily linear) function of the solution $u(p)$. As an example, we may consider $\Phi(u)$ as the average value of u over the entire domain Ω. In optimisation, ϕ is typically considered as a *target function* that needs to be minimised (or maximised) with respect to a deterministic parameter tuple p. In uncertainty quantification, the parameters are assumed to be random variables subject to a (given) probability distribution. One then aims at deriving stochastic properties of the *quantity of interest* ϕ from the stochastic assumptions on the input data.

Both in the deterministic and the stochastic case, the treatment of the parameter-dependent problem (1) faces two major difficulties. On the one hand, any instance of $p \in P$ requires the solution of a (typically infinite-dimensional) deterministic problem (1) which needs to be approximated numerically. On the other hand, the number of deterministic problems to be solved quickly increases with the dimension d of the parameter space P. This requires the development of special techniques that can cope even with high parameter dimensions d.

1.1 Tensor-structured Approach

In this paper, we propose a new approach for the explicit approximation of ϕ from (2) from a relatively small and carefully chosen set of samples $p \in P$. To this end, we first assume that the parameter space P can be represented as the cartesian product

$$P = P_1 \times \ldots \times P_d \tag{3}$$

with bounded intervals $P_\mu \subset \mathbb{R}$. In a stochastic context, such a setting arises if the random variables $p_\mu \in P_\mu$, $\mu = 1, \ldots, d$, are independent. If this independence is not given a priori, it may result from a truncated Karhunen-Loève expansion [22].

As a next step, we consider a discretisation of P by a tensor grid of n_μ points in each direction $\mu = 1, \ldots, d$ given by $\xi_{\mu,i_\mu} \in P_\mu$, $i_\mu = 1, \ldots, n_\mu$. Such a discretisation may, e.g., arise from a multivariate interpolation scheme or from a piecewise constant approximation of $p_\mu \in P_\mu$. The values of ϕ on the tensor grid can be represented by a d-dimensional tensor $A \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ defined by

$$A_{(i_1,\ldots,i_d)} := \phi(\xi_{1,i_1}, \ldots, \xi_{d,i_d}).$$

A full representation of A in terms of all its entries quickly becomes intractable as the parameter dimension d increases. We therefore suggest to approximate A in a data-sparse format that does not suffer from an exponential complexity with respect to d.

A quite general framework for the low-parametric representation of tensors has been introduced in [17] which we further analysed in [14]. In the so-called *hierarchical tensor* (or *hierarchical Tucker*) *format*, a tensor is represented by a number of parameters that scales only linearly in the dimension d. Our aim is to adaptively construct an approximation of A in this format by the evaluation of a relatively small number of entries $A_{(i_1,\dots,i_d)}$. In [4], we have introduced a black box approach which is adaptive in the sense that it finds the most relevant information from A automatically in order to reach a given (heuristic) target accuracy ε.

Once an approximation $\tilde{A} \approx A$ in the hierarchical tensor format is available, the evaluation of any entry of $\tilde{A}$ becomes a cheap online task. This, e.g., allows for the fast online optimisation of parameters. Moreover, stochastic properties of ϕ with respect to a given probability distribution of the parameter vector p can be computed in a straightforward way without the need of solving the original PDE problem.

Low-rank approximations for a full discretisation of the parametrised system (1) by tensor techniques have also been studied in [19, 23, 12, 18, 11, 20, 30].

1.2 Alternative Solution Strategies

A natural approach for the solution of (1) is to consider u as an element of the tensor product space $H \otimes V$ where H is a suitably chosen Hilbert space on the spatial domain Ω, e.g. $H = H_0^1(\Omega)$, and V is a Hilbert space on the parameter domain, e.g. $V = L^2(P)$. One can then seek for approximations $u \approx u_{h,n} \in H_h \otimes V_n$ in finite-dimensional subspaces $H_h \subset H$, $V_n \subset V$. In many cases, H_h represents a standard finite-element space spanned by piecewise polynomials whereas V_n corresponds to a space spanned by global polynomials up to a certain degree.

In the stochastic Galerkin method [28, 29, 2], one employs a full Galerkin discretisation of (1) with respect to the space $H_h \otimes V_n$. This typically leads to a very large coupled system of linear equations that needs to be solved numerically. Alternatively, one can only consider a Galerkin discretisation of (1) with respect to H_h and construct an approximation $u_{h,n} \in H_h \otimes V_n$ from suitably chosen collocation points $\xi_i \in P$, $i = 1, \dots, N$, in the form $u_{h,n}(p) = \sum_{i=1}^{N} u_h(\xi_i) v_i(p)$ with $u_h \in H_h$, $v_i \in V_n$, cf. [1, 8]. In this case, each collocation point ξ_i gives rise to a linear system with respect to the spatial discretisation $H_h \subset H$ which can be solved independently for all $i = 1, \dots, N$.

If the space V is discretised by a tensor-product polynomial space V_n of order n in each direction, both the stochastic Galerkin and the stochastic collocation method suffer from an exponential growth of the dimension of V_n with respect to d. If the solution u possesses certain regularity properties with respect to p, one can neglect a large number of higher-order terms which has inspired the development of sparse grid techniques [6, 9].

Similar to the stochastic collocation approach, reduced basis methods [27, 8] rely on the evaluation of the solution u at suitably chosen samples $\xi_i \in P$, $i = 1, \dots, M$. They are based on the observation that in many applications the set of all solutions $\{u(p) : p \in P\}$ can be well approximated by its projection to a finite and low-dimensional vector space. In particular, this means that there needs to exist a low-rank approximation of the form $u(p) \approx \sum_{i=1}^{M} u(\xi_i) v_i(p)$ with $v_i \in V_n \subset V$. The subspace V_n does not necessarily have to be spanned by polynomials but is often constructed in a greedy way such that one has $M \ll N$. As a crucial ingredient, an efficient offline-online strategy requires the *affine* parameter dependence of problem (1), i.e., there exists a representation of the form $\mathcal{D}(u,p) = \sum_{j=1}^{Q} \Theta_j(p) \mathcal{D}_j(u)$. Extensions to the non-affine case have been studied in [15, 25].

In a stochastic context, it is often sufficient to compute expected values of the output ϕ from (2). The corresponding high-dimensional integrals can then be approximated by Monte Carlo, quasi-Monte Carlo, or multilevel Monte Carlo techniques [10, 21, 7]. An alternative randomised technique has been proposed in [24].

2 A Model Problem

For a bounded Lipschitz domain $\Omega \subset \mathbb{R}^\ell$ we consider the diffusion problem

$$\begin{aligned} -\operatorname{div}(a(x,p)\nabla u(x,p)) &= f(x,p), \quad && x \in \Omega, \\ u(x,p) &= 0, && x \in \partial\Omega, \end{aligned} \tag{4}$$

for all $p \in P \subset \mathbb{R}^d$. Given $f \in H^{-1}(\Omega)$ and a uniformly elliptic diffusion coefficient a satisfying

$$0 < a_{\min} \leq a(x,p) \leq a_{\max} < \infty, \quad x \in \Omega,\, p \in P,$$

problem (4) is well-posed such that there exists a unique solution $u(\cdot,p) \in H_0^1(\Omega)$ for all $p \in P$. For ease of presentation we omit a possible parameter dependency of the boundary $\Omega = \Omega(p)$. However, our method identically is applicable to this problem and it does not require any modification or adaptation.

A typical quantity of interest might be given by the average value of the solution u over the entire domain Ω, i.e.,

$$\phi(p) = \Phi_{\text{avg}}(u(p)) := \frac{1}{|\Omega|}\int_\Omega u(x,p)\,\mathrm{d}x, \quad p \in P.$$

Since each evaluation of ϕ at a parameter $p \in P$ requires the possibly costly (approximate) solution of (4), our aim is to construct an approximation $\tilde{\phi} \approx \phi$ that can be evaluated efficiently. Assuming the cartesian product structure (3) of P, a natural approach is to seek for approximations $\tilde{\phi}$ of the form

$$\tilde{\phi}(p) := \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} A_{(i_1,\ldots,i_d)} \prod_{\mu=1}^{d} \hat{\phi}_{\mu,i_\mu}(p_\mu) \tag{5}$$

with univariate functions $\hat{\phi}_{\mu,i_\mu} : P_\mu \to \mathbb{R}$ for $\mu = 1,\ldots,d$ with suitable coefficients $A_{(i_1,\ldots,i_d)} \in \mathbb{R}$. Approximations of this form might, e.g., arise from a multivariate interpolation scheme or from a piecewise constant approximation of ϕ. In this case, the coefficients $A_{(i_1,\ldots,i_d)} \in \mathbb{R}$ are usually determined by the values of ϕ on a tensor grid given by

$$A_{(i_1,\ldots,i_d)} := \phi(\xi_{1,i_1},\ldots,\xi_{d,i_d})$$

with points $\xi_{\mu,i_\mu} \in P_\mu$, $\mu = 1,\ldots,d$.

For $d \gg 1$, an approximation of the form (5) will only lead to an efficient scheme if the coefficients $A_{(i_1,\ldots,i_d)}$ can (approximately) be represented in a data-sparse way.

3 Hierarchical Tensor Format

Given $d \in \mathbb{N}$, let $\mathcal{I}_1, \dots, \mathcal{I}_d$ be finite index sets with $\mathcal{I} := \mathcal{I}_1 \times \cdots \times \mathcal{I}_d$. Already for moderate d, the data-sparse representation of tensors $A \in \mathbb{R}^{\mathcal{I}}$ is of interest. A quite general concept for the representation of tensors has been introduced in [17] which we further analysed in [14]. In this approach, data-sparsity is obtained by a hierarchical representation of tensors which can be applied even in high dimensions d. Independently, a similar concept has been developed in [26].

As a first important ingredient, we define a matrix representation of a given tensor.

Definition 1 (matricisation)**.** Let $D := \{1, \dots, d\}$. Given a subset $t \subset D$ with complement $[t] := D \setminus t$, the *matricisation*

$$\mathcal{M}_t : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{I}_t} \otimes \mathbb{R}^{\mathcal{I}_{[t]}}, \qquad \mathcal{I}_t := \bigtimes_{\mu \in t} \mathcal{I}_\mu,\, \mathcal{I}_{[t]} := \bigtimes_{\mu \in [t]} \mathcal{I}_\mu,$$

of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ is defined by its entries

$$\mathcal{M}_t(A)_{(i_\mu)_{\mu \in t}, (i_\mu)_{\mu \in [t]}} := A_{(i_1, \dots, i_d)}, \qquad i_\mu \in \mathcal{I}_\mu,\, \mu \in D.$$

In order to allow for a hierarchical representation, the subsets $t \subset D$ are organised in a binary tree.

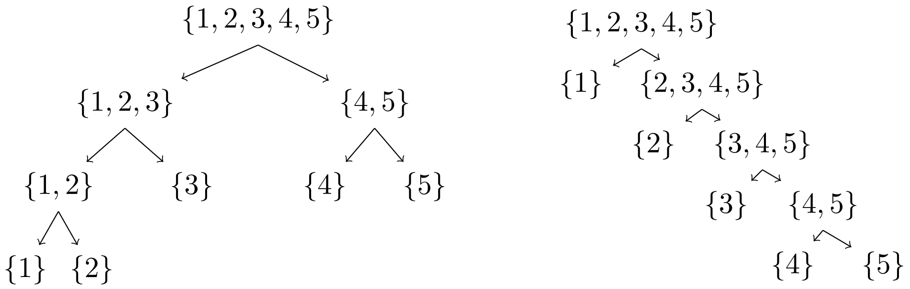


Figure 1: Left: A typical balanced binary dimension tree for $d = 5$. Right: A maximal unbalanced dimension tree as it is used for the TT (tensor train) or MPS (matrix product states) format.

Definition 2 (dimension tree)**.** Let $D := \{1, \dots, d\}$. A tree T_D is called a *dimension tree*, cf. Figure 1, if the following three conditions hold:

(a) the index set D is the root of the tree T_D,

(b) all vertices $t \in T_D$ are non-empty subsets $t \subset D$,

(c) every vertex $t \in T_D$ with $\#t \geq 2$ has two sons $t_1, t_2 \in T_D$ with the property

$$t = t_1 \cup t_2, \qquad t_1 \cap t_2 = \emptyset.$$

The set of leaves of T_D is defined by $\mathcal{L}(T_D) := \{t \in T_D : \#t = 1\}$. For all $t \in T_D \setminus \mathcal{L}(T_D)$, we denote the set of sons of t by $\mathrm{sons}(t)$.

Based on the concept of the matricisation of tensors and the definition of a dimension tree, we can now introduce the hierarchical tensor format.

Definition 3 (hierarchical rank, hierarchical format)**.** Let T_D be a dimension tree. The *hierarchical rank* $\boldsymbol{k} := (k_t)_{t \in T_D}$ of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ is defined by

$$k_t := \mathrm{rank}(\mathcal{M}_t(A)), \quad t \in T_D.$$

For a given hierarchical rank $\boldsymbol{k} := (k_t)_{t \in T_D}$, the *hierarchical format* $\mathcal{H}_{\boldsymbol{k}}$ is defined by

$$\mathcal{H}_{\boldsymbol{k}} := \{A \in \mathbb{R}^{\mathcal{I}} : \mathrm{rank}(\mathcal{M}_t(A)) \leq k_t,\, t \in T_D\}.$$

A key feature of a tensor in $\mathcal{H}_{\boldsymbol{k}}$ is that it can be represented in a recursive fashion.

Lemma 4 (cf. [14])**.** *Let $A \in \mathcal{H}_{\boldsymbol{k}}$. Then $A = (U_D)_{\cdot,1}$ can be represented by the recursive relation*

$$(U_t)_{\cdot,j} = \sum_{j_1=1}^{k_{t_1}} \sum_{j_2=1}^{k_{t_2}} (B_t)_{j,j_1,j_2} (U_{t_1})_{\cdot,j_1} \otimes (U_{t_2})_{\cdot,j_2}, \qquad j = 1, \ldots, k_t \tag{6}$$

for all $t \in T_D \setminus \mathcal{L}(T_D)$ with $\mathrm{sons}(t) = \{t_1, t_2\}$ *where $B_t \in \mathbb{R}^{k_t \times k_{t_1} \times k_{t_2}}$ and $U_t \in \mathbb{R}^{\mathcal{I}_t \times k_t}$, $\mathcal{I}_t := \times_{\mu \in t} \mathcal{I}_\mu$, for all $t \in T_D$.*

As a consequence of the last lemma, one only needs to store the matrices $U_t \in \mathbb{R}^{\mathcal{I}_\mu \times k_t}$ in the leaves $t = \{\mu\} \in \mathcal{L}(T_D)$ and the transfer tensors $B_t \in \mathbb{R}^{k_t \times k_{t_1} \times k_{t_2}}$ for all inner nodes $t = \{t_1, t_2\} \in T_D \setminus \mathcal{L}(T_D)$ in order to represent a tensor in $\mathcal{H}_{\boldsymbol{k}}$. The complexity for this representation then sums up to $\mathcal{O}(dk^3 + dkn)$, where $k := \max_{t \in T_D} k_t$, $n := \max_{\mu \in D} \#\mathcal{I}_\mu$.

The hierarchical rank of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ can be computed by standard linear algebra tools. Moreover, we have developed a truncation procedure [14] that computes an approximation of the best approximation of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ in $\mathcal{H}_{\boldsymbol{k}}$. This hierarchical singular value decomposition ($\mathcal{H}$-SVD) yields a tensor $A_{\mathcal{H}\text{-SVD}} \in \mathcal{H}_{\boldsymbol{k}}$ with the property that

$$\|A - A_{\mathcal{H}\text{-SVD}}\|_2 \leq \sqrt{2d-3} \min_{A_{\mathrm{best}} \in \mathcal{H}_{\boldsymbol{k}}} \|A - A_{\mathrm{best}}\|_2 \,.$$

If the input tensor A is already given in hierarchical format, i.e. $A \in \mathcal{H}_{\boldsymbol{k}}$, the $\mathcal{H}$-SVD can be computed in

$$\mathcal{O}(dk^4 + dnk^2).$$

Similar results have been obtained for the TT format in [26].

A straight-forward approach to solve systems with vectors and operators in the hierarchical format is as follows: use a standard iterative solver and project after each step the iterates back to some small or adaptively determined hierarchical rank via the $\mathcal{H}$-SVD. If the adaptive determination of the rank is done correctly, then such an approach can guarantee convergence as for the iterative solver without projection. However, the rank might increase unfavourably during the iteration. This is unavoidable, because even the most data-sparse accurate approximation of each exact iterate may require arbitrarily large ranks. Moreover, such an approach requires the iterative solver to work with structured vectors. It is therefore intrusive in the sense that the PDE solvers have to be modified down to the basic linear algebraic operations, again something unfavourable. What we propose in the following is a different approach where the basic building block is the mapping from a single parameter tuple p to a single output quantity

$$p \mapsto \Phi(u(p)).$$

4 Black Box Low-Rank Approximation

In [4] we have introduced an adaptive strategy for the approximation of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ in $\mathcal{H}_{\boldsymbol{k}}$ from the evaluation of a carefully chosen subset of its entries $A_{(i_1,\dots,i_d)}$. The main idea is to recursively construct low-rank approximations of matrices in an adaptive way.

4.1 Cross approximation for matrices

The approximation of a matrix $M \in \mathbb{R}^{\mathcal{I}_1\times\mathcal{I}_2}$ by the outer product of particular rows and columns of M has been analysed in [13]. The main theorem is as follows.

Theorem 5 ([13, Corollary 3.1]). *Let $M \in \mathbb{R}^{\mathcal{I}_1\times\mathcal{I}_2}$. If there exists a matrix $B \in \mathbb{R}^{\mathcal{I}_1\times\mathcal{I}_2}$ with the property*

$$\|M - B\|_2 \le \varepsilon, \qquad \operatorname{rank}(B) \le k,$$

then there exist a subset $\mathcal{P} \subset \mathcal{I}_1$ of row indices and a subset $\mathcal{Q} \subset \mathcal{I}_2$ of column indices with $\#\mathcal{P} = \#\mathcal{Q} \le k$ and a matrix $S \in \mathbb{R}^{\mathcal{P}\times\mathcal{Q}}$ such that

$$\tilde{M} := M|_{\mathcal{I}_1\times\mathcal{Q}} \cdot S^{-1} \cdot M|_{\mathcal{P}\times\mathcal{I}_2}$$

approximates M with an error of

$$\left\|M - \tilde{M}\right\|_2 \le \varepsilon\left(1 + 2\sqrt{k}\left(\sqrt{\#\mathcal{I}_1} + \sqrt{\#\mathcal{I}_2}\right)\right).$$

A practical construction which is based on successive rank one approximations has been introduced in [5]. Starting with $X^0 := 0$, the idea is to construct approximations $X^j \approx M$, $j = 1,\dots,k$, by

$$X^j := X^{j-1} + R^{j-1}|_{\mathcal{I}_1\times\{q_j\}} \cdot (R^{j-1}_{p_j,q_j})^{-1} \cdot R^{j-1}|_{\{p_j\}\times\mathcal{I}_2}, \qquad R^{j-1} := M - X^{j-1}, \tag{7}$$

where $p_j \in \mathcal{I}_1$, $q_j \in \mathcal{I}_2$ are appropriately chosen pivot elements. Using the notation of Theorem 5, we have that

$$S = M|_{\mathcal{P}\times\mathcal{Q}}, \quad \mathcal{P} := \{p_1,\dots,p_k\}, \quad \mathcal{Q} := \{q_1,\dots,q_k\}.$$

The construction (7) is adaptive in the sense that the size $|R^{j-1}_{p_j,q_j}|$ gives an estimate for the norm $\|M - X^{j-1}\|_\infty$ of the remainder.

4.2 Generalised cross approximation for higher-order tensors

Let $A \in \mathbb{R}^{\mathcal{I}}$ and let T_D be a dimension tree. Given $t \in T_D$, the cross approximation technique for matrices can in principle be used to construct low-rank approximations of the matricisations $M := \mathcal{M}_t(A) \in \mathbb{R}^{\mathcal{I}_t \times \mathcal{I}_{[t]}}$ of the form

$$M \approx M|_{\mathcal{I}_t \times \mathcal{Q}_t} \cdot M|_{\mathcal{P}_t \times \mathcal{Q}_t}^{-1} \cdot M|_{\mathcal{P}_t \times \mathcal{I}_{[t]}} \tag{8}$$

with pivot sets $\mathcal{P}_t \subset \mathcal{I}_t$, $\mathcal{Q}_t \subset \mathcal{I}_{[t]}$. Note, however, that the index sets $\mathcal{I}_t, \mathcal{I}_{[t]}$ are typically very large such that an explicit representation of the form (8) is intractable. We therefore propose a recursive strategy that directly constructs an approximation $\tilde{A} \approx A$ in $\mathcal{H}_{\boldsymbol{k}}$ from well chosen pivots $\mathcal{P}_t, \mathcal{Q}_t$ for all $t \in T_D$.

Assume for a moment that for each node $t \in T_D$ of the dimension tree and correspondingly for each matricisation $\mathcal{M}_t(A)$ we already have selected pivot sets $\mathcal{P}_t, \mathcal{Q}_t$ where we enforce a nestedness condition of the form

$$\mathcal{Q}_s \subset \mathcal{I}_{[t]} \times \mathcal{Q}_t, \qquad s \in \mathrm{sons}(t), \tag{9}$$

cf. [4]. We can then use the nestedness property (6) for nodes t with $\mathrm{sons}(t) = \{t_1, t_2\}$ and define the (approximate) transfer tensor $\tilde{B}_t$ by

$$(\tilde{B}_t)_{j,\cdot,\cdot} := S_{t_1}^{-1} M_{t,j} S_{t_2}^{-\top} \tag{10}$$

where the coupling matrix $M_{t,j}$ is defined by

$$(M_{t,j})_{p,q} := \mathcal{M}_t(A)_{(p,q),j}, \qquad p \in \mathcal{P}_{t_1} \subset \mathcal{I}_{t_1}, q \in \mathcal{P}_{t_2} \subset \mathcal{I}_{t_2}, j \in \mathcal{P}_t \subset \mathcal{I}_t$$

and the two matrices S_{t_1}, S_{t_2} are

$$S_{t_1} := \mathcal{M}_{t_1}(A)|_{\mathcal{P}_{t_1} \times \mathcal{Q}_{t_1}}, \quad S_{t_2} := \mathcal{M}_{t_2}(A)|_{\mathcal{P}_{t_2} \times \mathcal{Q}_{t_2}}.$$

For the leaves we have to specify the (approximate) $\tilde{U}_t$, which are given by

$$\tilde{U}_t := \mathcal{M}_t(A)|_{\mathcal{I}_t \times \mathcal{Q}_t}.$$

A necessary assumption here is that for every node $t \in T_D$ the submatrix S_t is invertible, which can easily be enforced by a rank revealing decomposition and subsequent reduction of the pivot sets. A particularly interesting fact is already stated in [4]: If all submatrices S_t are of full rank

$$\mathrm{rank}(S_t) = k_t = \mathrm{rank}(\mathcal{M}_t(A)),$$

then the tensor defined by $\tilde{B}, \tilde{U}$ is the same as the original tensor:

$$\tilde{A} = A.$$

An algorithmic description of this recursive strategy is given in Algorithm 1 where the first call with $t = D$ reads

$$\texttt{rec\_cross\_approx}(D, \emptyset).$$

In Algorithm 1 there are still three ingredients to be specified:

Algorithm 1 rec_cross_approx($t, \mathcal{Q}_t$)

```
if $t \in T_D \setminus \mathcal{L}(T_D)$ then
    $\mathrm{sons}(t) = \{t_1, t_2\}$
    $(\mathcal{P}_{t_1}, \mathcal{Q}_{t_1}, S_{t_1}, k_{t_1})$ = find_pivots($t_1, \mathcal{Q}_t$)
    $(\mathcal{P}_{t_2}, \mathcal{Q}_{t_2}, S_{t_2}, k_{t_2})$ = find_pivots($t_2, \mathcal{Q}_t$)
    $(\tilde{B}_t)_{j,\cdot,\cdot} := S_{t_1}^{-1} M_{t,j} S_{t_2}^{-\top}$ as in (10)
    rec_cross_approx($t_1, \mathcal{Q}_{t_1}$)
    rec_cross_approx($t_2, \mathcal{Q}_{t_2}$)
else
    $\tilde{U}_t := \mathcal{M}_t(A)|_{\mathcal{I}_t \times \mathcal{Q}_t}$
end if
```

1. the dimension tree T_D,
2. the rank (or rank bound) k_t for each node $t \in T_D$, and
3. the pivot sets $\mathcal{P}_t, \mathcal{Q}_t$ for each node $t \in T_D$.

The choice of the pivots $\mathcal{P}_t, \mathcal{Q}_t$ by the still abstract routine find_pivots is the crucial part of the algorithm. In [4] this is done by a search along fibres of the tensor. This strategy is reasonable if the evaluation of entries of the tensor is negligible, e.g. the evaluation of a simple function like $f(x) = 1/\|x\|$. In the PDE scenario considered here the situation is entirely different. We want to minimize the number of sample points $p \in P$ because each sample requires the solution of a possibly complicated PDE. On the other hand, the storage complexity for the compressed tensor as well as arithmetic operations in the tensor format are entirely irrelevant. Therefore we have to modify the choice of the pivot sets and our sampling strategy as outlined in the forthcoming subsection.

As mentioned above, both the dimension tree as well as the nodewise ranks k_t have to be determined. For some applications the choice of the tree might be straight-forward, or even irrelevant: if the tensor can be approximated in the CP-format with rank K, then any dimension tree is sufficient and the nodewise ranks can be bounded by K. If this is not the case, then a heuristic approach is required to find a useful tree. In [3] we present an agglomerative procedure to obtain the tree T_D. There we also specify how the nodewise ranks can be estimated by sampling of random submatrices. From now on we assume that the dimension tree T_D is given, as well as a rank bound K_t for each node $t \in T_D$.

4.3 Partial Random Sampling

Due to the nestedness requirement (9), we have to choose the pivot sets in an ordered way descending from the root, level by level, down to the leaves. For each node we modify the pivot selection from [4] as follows:

Assume we have constructed the pivots $\mathcal{P}_t, \mathcal{Q}_t$ for a node $t \in T_D \setminus \mathcal{L}(T_D)$. For a son $s \in \mathrm{sons}(t)$ we then proceed by the strategy indicated in Algorithm 2.

Algorithm 2 $(\mathcal{P}_s, \mathcal{Q}_s, S_s, k_s) = \texttt{find\_pivots}(s, \mathcal{Q}_t)$

```
1: choose (at random) index sets $\hat{P} \subset \mathcal{I}_s$ and $\hat{Q} \subset \mathcal{I}_{[t]} \times \mathcal{Q}_t$ with $\#\hat{P}, \#\hat{Q} = K_t$ for a given rank bound $K_t \in \mathbb{N}$
2: define the (random) submatrix $\hat{M} := \mathcal{M}_s(A)|_{\hat{P}\times\hat{Q}}$
3: determine subsets $\mathcal{P}_s \subset \hat{P}$, $\mathcal{Q}_s \subset \hat{Q}$ with $k_s := \#\mathcal{P}_s = \#\mathcal{Q}_s$ such that $S_s := \hat{M}|_{\mathcal{P}_s\times\mathcal{Q}_s}$ is of almost maximal determinant
```

As long as the number K_t of random indices is not too large, we can afford to approximate the submatrix $\hat{M}$ by some standard rank revealing decomposition for matrices, as, e.g., the cross approximation technique from Section 4.1. In this case, the number of entries required from A in Algorithm 2 is bounded by $2k_tK_t$ for each node $t \in T_D \setminus \mathcal{L}(T_D)$. For the construction of the matrices $M_{t,j}$ from (10) in Algorithm 1, one additionally needs to compute $k_tk_{t_1}k_{t_2}$ entries, where $\mathrm{sons}(t) = \{t_1, t_2\}$. At the leaves $t \in \mathcal{L}(T_D)$, the evaluation of the matrices $\tilde{U}_t$ requires at most k_tn evaluations, where $n := \max_{\mu=1,\dots,d} \#\mathcal{I}_\mu$. Since a dimension tree has at most d leaf and d non-leaf nodes, we end up with

$$N_{\mathrm{eval}} = \mathcal{O}\left(dk^3 + dkK + dkn\right), \qquad k := \max_{t\in T_D} k_t, \qquad K := \max_{t\in T_D} K_t,$$

entries that have to be sampled.

Provided that $K = \mathcal{O}(k^2)$, this already corresponds to the storage complexity of a tensor in $\mathcal{H}_{\boldsymbol{k}}$, i.e., the sampling strategy would be optimal if the approximation quality of $\tilde{A}$ is close to the best possible approximation of A with the same data-sparsity. We provide evidence that this is indeed true in the numerical examples section.

5 Applications

Once a data-sparse representation of A in the hierarchical format $\mathcal{H}_{\boldsymbol{k}}$ has been found, the evaluation of a single entry $A_{(i_1,\dots,i_d)}$ requires only $\mathcal{O}(dk^3)$ operations, cf. [16]. Moreover, the hierarchical representation can be exploited to efficiently compute a posteriori quantities of

$$\tilde{\phi}(p) := \sum_{i_1\in\mathcal{I}_1} \cdots \sum_{i_d\in\mathcal{I}_d} A_{(i_1,\dots,i_d)} \prod_{\mu=1}^{d} \hat{\phi}_{\mu,i_\mu}(p_\mu)$$

from (5).

5.1 A posteriori Evaluation of Statistics

In uncertainty quantification, a typical aim is to estimate stochastic properties of $\phi : P \to \mathbb{R}$ for a given probability distribution of the parameters $p \in P$. Let $\pi : P \to \mathbb{R}$ be a known probability density function associated to the parameters $p \in P$. We then look for the expected value

$$\mathbb{E}[\phi] = \int_P \phi(p)\pi(p)\,\mathrm{d}p. \tag{11}$$

If the parameters $p_1, \dots, p_d$ are independent, we can express π by a separable representation of the form

$$\pi(p) = \pi_1(p_1) \cdot \ldots \cdot \pi_d(p_d), \quad p_\mu \in P_\mu,$$

with $\pi_\mu : P_\mu \to \mathbb{R}$ for $\mu = 1, \dots, d$.

Assume now that $A \in \mathcal{H}_{\boldsymbol{k}}$ such that A can be represented by transfer tensors $B_t \in \mathbb{R}^{k_t \times k_{t_1} \times k_{t_2}}$ for all $t \in T_D \setminus \mathcal{L}(T_D)$ and matrices $U_t \in \mathbb{R}^{\mathcal{I}_\mu \times k_\mu}$ for all $t = \{\mu\} \in \mathcal{L}(T_D)$. An approximation to $\mathbb{E}[\phi]$ from (11) can then be computed by a simple Euclidean inner product between A and an elementary tensor $w = w_1 \otimes \ldots \otimes w_d$ with $w_\mu \in \mathbb{R}^{\mathcal{I}_\mu}$ for $\mu = 1, \dots, d$.

For all $\mu = 1, \dots, d$, one first calculates the vectors $w_\mu \in \mathbb{R}^{\mathcal{I}_\mu}$ by

$$(w_\mu)_{i_\mu} := \int_{P_\mu} \hat{\phi}_{\mu, i_\mu}(p_\mu) \pi_\mu(p_\mu) \, \mathrm{d}p_\mu, \quad i_\mu \in \mathcal{I}_\mu.$$

For all leaves $t = \{\mu\} \in \mathcal{L}(T_D)$, one then computes $v_t \in \mathbb{R}^{k_t}$ by

$$(v_t)_j := \sum_{i_\mu \in \mathcal{I}_\mu} (U_t)_{i_\mu, j} (w_\mu)_{i_\mu} \qquad j = 1, \dots, k_t,$$

i.e., $v_t = U_t^\top w_\mu$. For all inner nodes $t \in T_D \setminus \mathcal{L}(T_D)$ with $\mathrm{sons}(t) = \{t_1, t_2\}$, one defines $v_t \in \mathbb{R}^{k_t}$ by

$$(v_t)_j := \sum_{j_1=1}^{k_{t_1}} \sum_{j_2=1}^{k_{t_2}} (B_t)_{j, j_1, j_2} (v_{t_1})_{j_1} (v_{t_2})_{j_2}, \qquad j = 1, \dots, k_t.$$

The final approximation is then given by

$$\mathbb{E}[\phi] \approx \mathbb{E}[\tilde{\phi}] = \langle A, w \rangle = (v_D)_1.$$

This scheme can be carried out in $\mathcal{O}(dk^3 + dkn)$ operations, $k := \max_{t \in T_D} k_t$, $n := \max_{\mu = 1, \dots, d} \# \mathcal{I}_\mu$.

5.2 Explicit Representation of Gradients

In optimisation, a typical goal is to find minima of ϕ, i.e., one seeks $p^* \in P$ s.t.

$$\phi(p^*) = \min_{p \in P} \phi(p).$$

Whereas a global minimisation is in general hard to perform, local minimisers can, e.g., be found by gradient-based methods. These techniques require the computation of $\nabla_p \phi : \mathbb{R}^d \to \mathbb{R}^d$ at certain points $q = (q_1, \dots, q_d) \in P$. Due to (5), one needs to calculate

$$\frac{\partial \phi}{\partial p_\nu}(q) = \sum_{i_1 \in \mathcal{I}_1} \cdots \sum_{i_d \in \mathcal{I}_d} A_{(i_1, \dots, i_d)} \frac{\partial \hat{\phi}_{\nu, i_\nu}}{\partial p_\nu}(q_\nu) \prod_{\substack{\mu=1 \\ \mu \neq \nu}}^{d} \hat{\phi}_{\mu, i_\mu}(q_\mu), \qquad \nu = 1, \dots, d.$$

Given $A \in \mathcal{H}_{\boldsymbol{k}}$, one can evaluate $\partial \phi / \partial p_\nu$ at $q \in P$ by a similar recursion as above in a complexity of $\mathcal{O}(dk^3 + dkn)$ operations.

6 Numerical Examples

Given a domain $\Omega \subset \mathbb{R}^2$ and a parameter space $P \subset \mathbb{R}^d$, we consider the parametric diffusion problem

$$-\operatorname{div}(a(x,p)\nabla u(x,p)) = f(x,p), \quad x \in \Omega,\, p \in P, \tag{12}$$

completed with suitable boundary conditions. We assume that $P = P_1 \times \ldots \times P_d$ is discretised by a tensor grid constructed from Chebyshev points $\xi_{\mu,i_\mu} \in P_\mu$, $i_\mu = 1,\ldots,n_\mu$, in each direction $\mu = 1,\ldots,d$. For simplicity, let $n := n_1 = \ldots = n_d$ such that $\mathcal{I} := \{1,\ldots,n\}^d$. Given an output $\phi : P \to \mathbb{R}$, we want to approximate $A \in \mathbb{R}^{\mathcal{I}}$ defined by

$$A_{(i_1,\ldots,i_d)} := \phi(\xi_{1,i_1},\ldots,\xi_{d,i_d}).$$

In the numerical tests, we are interested in the following questions:

1. Does the tensor $A \in \mathbb{R}^{\mathcal{I}}$ admit a data-sparse approximation $\tilde{A} \in \mathcal{H}_{\boldsymbol{k}}$? How does the storage complexity of $\tilde{A}$ depend on a given accuracy?

2. How many entries of A (i.e, deterministic solutions of the PDE) are required to construct $\tilde{A}$?

We study both questions for two typical examples taken from the literature. As a reference domain, we choose $\Omega = (0,1)^2$ discretised by an equidistant grid of 128 points in both spatial directions. We then apply a finite-element discretisation with 16384 linear elements to solve (12) numerically for any $p \in P$. We restrict ourselves to a parameter space of $P = [\frac{1}{2}, 2]^d$ discretised by $n = 10$ Chebyshev points in each direction $\mu = 1,\ldots,d$.

For a prescribed error tolerance $\varepsilon > 0$, we construct an approximation $\tilde{A} \approx A$ in $\mathcal{H}_{\boldsymbol{k}}$ by the black box approach described in Section 4.2. In an a posteriori calculation we estimate the relative error $\|A - \tilde{A}\|_{2,\mathcal{J}} / \|A\|_{2,\mathcal{J}}$ with

$$\|X\|_{2,\mathcal{J}} := \left(\sum_{\boldsymbol{i} \in \mathcal{J}} X_{\boldsymbol{i}}^2 \right)^{1/2}, \quad X \in \mathbb{R}^{\mathcal{I}},$$

for a randomly chosen test index set $\mathcal{J} \subset \mathcal{I}$ of size $\#\mathcal{J} = 100$. The number of required entries from A is denoted by N_{eval}.

To measure the storage requirements of $\tilde{A}$ in $\mathcal{H}_{\boldsymbol{k}}$, we compute the effective rank k_{eff} of $\tilde{A}$ which corresponds to a storage complexity of $\mathcal{O}((d-1)k_{\text{eff}}^3 + dnk_{\text{eff}})$. This is done directly after the black box approximation ($k_{\text{eff}}^{\text{bb}}$) and once again after a truncation of $\tilde{A}$ with accuracy ε ($k_{\text{eff}}^{\text{trunc}}$) which removes redundancy from the hierarchical representation. Finally, we compare the number of tensor evaluations N_{eval} to the storage requirements $\operatorname{stor}(\tilde{A})$ of the compressed representation of $\tilde{A}$ characterised by $k_{\text{eff}}^{\text{trunc}}$.

6.1 Cookie Problem

As a first example, we consider the cookie problem (cf. [30])

$$\begin{aligned} -\operatorname{div}(a(x,p)\nabla u(x,p)) &= 1, \qquad x \in \Omega, \\ u(x,p) &= 0, \qquad x \in \partial\Omega. \end{aligned} \tag{13}$$

Given $m \in \mathbb{N}$, the domain Ω encloses disks $\Omega_{s,t} \subset \Omega$ of the form

$$\Omega_{s,t} := \{x \in \mathbb{R}^2 : \|x - x_{s,t}\|_2 \le r\}$$

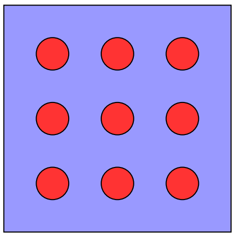
with radius $r := 1/(4m+2)$ and center $x_{s,t} := 4r(s,t)^\top - r(1,1)^\top$ for all $s,t = 1,\dots,m$. On each subdomain $\Omega_{s,t}$, the diffusion coefficient is supposed to be constant with

$$a(x,p) := \begin{cases} p_\mu, & x \in \Omega_{s,t},\ \mu := m(t-1)+s, \\ 1, & \text{otherwise.} \end{cases}$$

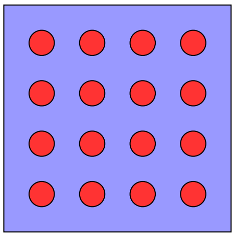
This results in a parameter space P of dimension $d = m^2$. An example for $d = 9$ and $d = 16$ is depicted in Figure 2. As an output quantity, we consider the average of the solution u over the entire domain, i.e.,

$$\phi(p) := \int_\Omega u(x,p)\,\mathrm{d}x, \quad p \in P. \tag{14}$$

The results of the tensor approximation can be found in Figure 2.



ε	$\frac{\|A-\tilde{A}\|_{2,\mathcal{J}}}{\|A\|_{2,\mathcal{J}}}$	N_{eval}	$k_{\text{eff}}^{\text{bb}}$	$k_{\text{eff}}^{\text{trunc}}$	$\frac{N_{\text{eval}}}{\text{stor}(\tilde{A})}$
1e-03	3.27e-04	1548	2.0	1.0	15.8
1e-04	1.03e-04	2784	3.0	1.9	12.3
1e-05	1.48e-05	3224	3.4	2.2	11.5
1e-06	2.74e-06	5338	4.7	3.4	8.6
1e-07	1.88e-07	9475	5.9	4.7	7.7



ε	$\frac{\|A-\tilde{A}\|_{2,\mathcal{J}}}{\|A\|_{2,\mathcal{J}}}$	N_{eval}	$k_{\text{eff}}^{\text{bb}}$	$k_{\text{eff}}^{\text{trunc}}$	$\frac{N_{\text{eval}}}{\text{stor}(\tilde{A})}$
1e-03	3.98e-04	2959	2.0	1.0	16.9
1e-04	2.81e-04	5261	3.1	1.7	15.5
1e-05	1.27e-05	8320	4.1	3.0	9.5
1e-06	3.75e-06	12736	5.3	3.9	8.2
1e-07	3.12e-07	26010	7.5	6.1	5.8

Figure 2: Cookie problem. Top: $d = 9$. Bottom: $d = 16$. Properties of the tensor approximation of A derived from (14) for different accuracies ε.

We can see that in both cases the relative error is always close to the prescribed error tolerance ε. The number of tensor evaluations N_{eval} and the representation rank

$k_{\text{eff}}^{\text{bb}}$ increase only moderately with the accuracy. The rank $k_{\text{eff}}^{\text{trunc}}$ of the compressed hierarchical representation of $\tilde{A}$ is always close to the black box based representation rank $k_{\text{eff}}^{\text{bb}}$. As a consequence, the number of tensor evaluation tends to be a small multiple of the storage complexity of $\tilde{A}$ in $\mathcal{H}_{\boldsymbol{k}}$.

Once the tensor $\tilde{A} \in \mathcal{H}_{\boldsymbol{k}}$ is available, we can directly compute an approximation $\mathbb{E}[\tilde{\phi}]$ of the expected value $\mathbb{E}[\phi]$ from (11) for a given density $\pi(p) = \prod_{\mu=1}^{d} \pi_\mu(p_\mu)$. Here, we assume that each π_μ corresponds to a uniform distribution of p_μ in $P_\mu = [\frac{1}{2}, 2]$. By means of a highly accurate tensor approximation we can estimate the relative error $|\mathbb{E}[\tilde{\phi}] - \mathbb{E}[\phi]| / |\mathbb{E}[\phi]|$. The dependence of this error on the number of evaluations of ϕ is compared to a standard Monte Carlo method in Figure 3. We observe that the tensor-structured approach clearly outperforms the Monte Carlo based approximation.

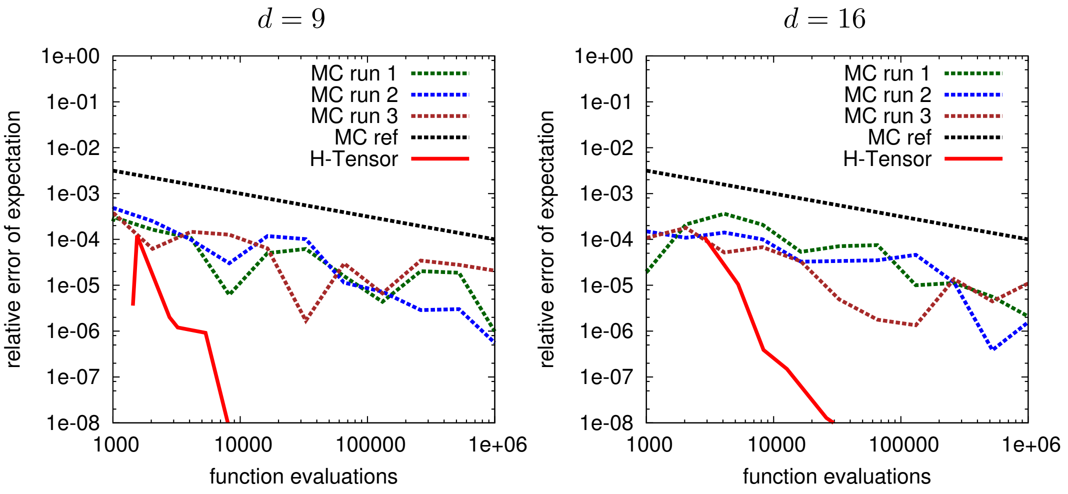


Figure 3: Cookie problem. Left: $d = 9$. Right: $d = 16$. Relative error in $\mathbb{E}[\phi]$ in dependence on the number of evaluations of ϕ.

6.2 Extended Cookie Problem

As a second example, we again consider the cookie problem (13) but now additionally allow a variation in the size of the cookies. Given $m \in \mathbb{N}$ and $r := 1/(4m+2)$, the domain Ω now encloses disks $\Omega_{s,t}(p) \subset \Omega$ of the form

$$\Omega_{s,t}(p) := \{x \in \mathbb{R}^2 : \|x - x_{s,t}\|_2 \leq r_{s,t}(p)\}$$

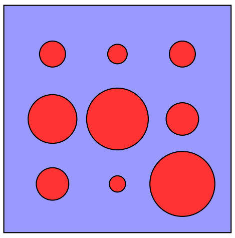
with center $x_{s,t} := 4r(s,t)^\top - r(1,1)^\top$ and parameter-dependent radius

$$r_{s,t}(p) := p_\mu r, \quad \mu := m(t-1) + s + m^2,$$

for all $s, t = 1, \ldots, m$. On each subdomain $\Omega_{s,t}(p)$, the diffusion coefficient is again supposed to be constant with

$$a(x,p) := \begin{cases} p_\mu, & x \in \Omega_{s,t}(p),\ \mu := m(t-1) + s, \\ 1, & \text{otherwise.} \end{cases}$$

This results in a parameter space P of dimension $d = 2m^2$. An example for $d = 18$ is given in Figure 4. As an output quantity, we again consider ϕ from (14). The results of the tensor approximation can be found in Figure 4.



ε	$\frac{\|A-\tilde{A}\|_{2,\mathcal{J}}}{\|A\|_{2,\mathcal{J}}}$	N_{eval}	$k_{\text{eff}}^{\text{bb}}$	$k_{\text{eff}}^{\text{trunc}}$	$\frac{N_{\text{eval}}}{\text{stor}(\tilde{A})}$
1e-03	1.69e-03	5625	2.9	1.7	14.7
1e-04	2.34e-04	7434	3.6	2.1	13.4
1e-05	4.95e-05	14271	5.3	3.8	8.9
1e-06	1.96e-05	39395	8.6	6.8	6.0
1e-07	2.73e-06	113691	14.1	12.0	3.6

Figure 4: Extended cookie problem for $d = 18$. Properties of the tensor approximation of A derived from (14) for different accuracies ε.

As in the previous example, the relative error is always close to the prescribed error tolerance ε. The representation rank $k_{\text{eff}}^{\text{bb}}$ is close to the rank $k_{\text{eff}}^{\text{trunc}}$ of the compressed representation of $\tilde{A}$ and increases only slowly with the accuracy. Moreover, the number of required tensor evaluations N_{eval} tends to be a small multiple of the storage complexity of $\tilde{A}$ in $\mathcal{H}_{\boldsymbol{k}}$.

6.3 Thermal Block Problem

As a third example, we consider the thermal block problem (cf. [27])

$$\begin{aligned} -\operatorname{div}(a(x,p)\nabla u(x,p)) &= 0, && x \in \Omega, \\ u(x,p) &= 0, && x \in \Gamma_{\text{top}}, \\ n(x)\cdot a(x,p)\nabla u(x,p) &= 0, && x \in \Gamma_{\text{left}} \cup \Gamma_{\text{right}}, \\ n(x)\cdot a(x,p)\nabla u(x,p) &= 1, && x \in \Gamma_{\text{bottom}}, \end{aligned}$$

with $\Gamma_{\text{bottom}} := [0,1]\times\{0\}$, $\Gamma_{\text{top}} := [0,1]\times\{1\}$, $\Gamma_{\text{left}} := \{0\}\times[0,1]$, $\Gamma_{\text{right}} := \{1\}\times[0,1]$.

Given $m \in \mathbb{N}$, the domain Ω is divided into subdomains $\Omega_{s,t} \subset \Omega$ of the form

$$\Omega_{s,t} := \left(\tfrac{s-1}{m}, \tfrac{s}{m}\right) \times \left(\tfrac{t-1}{m}, \tfrac{t}{m}\right)$$

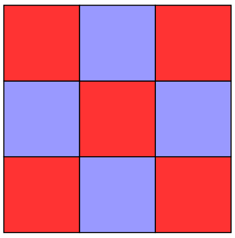
for all $s,t = 1,\ldots,m$. On each subdomain $\Omega_{s,t}$, the diffusion coefficient is supposed to be constant with

$$a(x,p) := p_\mu, \quad x \in \Omega_{s,t}, \;\; \mu := m(t-1)+s.$$

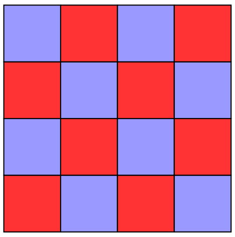
This results in a parameter space P of dimension $d = m^2$. An example for $d = 9$ and $d = 16$ is depicted in Figure 5. As an output quantity, we consider the average of the solution u over the bottom part of the boundary, i.e.,

$$\phi(p) := \int_{\Gamma_{\text{bottom}}} u(x,p)\,\mathrm{d}x, \quad p \in P. \tag{15}$$

The results of the tensor approximation can be found in Figure 5.



ε	$\frac{\lVert A-\tilde{A}\rVert_{2,\mathcal{J}}}{\lVert A\rVert_{2,\mathcal{J}}}$	N_{eval}	$k_{\text{eff}}^{\text{bb}}$	$k_{\text{eff}}^{\text{trunc}}$	$\frac{N_{\text{eval}}}{\text{stor}(\tilde{A})}$
1e-03	1.87e-03	5690	4.8	2.9	12.3
1e-04	3.79e-04	10819	6.4	5.0	7.6
1e-05	5.06e-05	25772	10.0	8.2	5.0
1e-06	5.25e-06	71422	15.2	12.9	3.9
1e-07	8.62e-07	145226	20.0	17.9	3.1



ε	$\frac{\lVert A-\tilde{A}\rVert_{2,\mathcal{J}}}{\lVert A\rVert_{2,\mathcal{J}}}$	N_{eval}	$k_{\text{eff}}^{\text{bb}}$	$k_{\text{eff}}^{\text{trunc}}$	$\frac{N_{\text{eval}}}{\text{stor}(\tilde{A})}$
1e-03	3.32e-03	8697	4.3	2.9	10.3
1e-04	7.27e-04	23945	7.1	5.9	5.8
1e-05	1.21e-04	79082	13.0	10.6	4.0
1e-06	1.05e-05	293840	22.1	18.7	2.9
1e-07	1.64e-06	1158435	37.8	32.5	2.2

Figure 5: Thermal block problem. Top: $d = 9$. Bottom: $d = 16$. Properties of the tensor approximation of A derived from (15) for different accuracies ε.

We can see that in both cases the relative error is always close to the prescribed error tolerance ε. The number of tensor evaluations N_{eval} and the representation rank $k_{\text{eff}}^{\text{bb}}$ increase significantly with the accuracy. This seems to be due to the fact that the parameters p_μ are geometrically much more coupled than in the cookie example from the previous section. As before, the rank $k_{\text{eff}}^{\text{trunc}}$ of the compressed hierarchical representation of $\tilde{A}$ is always close to the black box based representation rank $k_{\text{eff}}^{\text{bb}}$. As a consequence, the number of tensor evaluation tends to be a small multiple of the storage complexity of $\tilde{A}$ in $\mathcal{H}_{\boldsymbol{k}}$.

7 Conclusion

We have presented a non-intrusive sampling strategy for output quantities of parameter-dependent PDEs. The approximation is based on rank revealing factorisations of matricisations of a discrete representation of the parameter to output mapping. The underlying assumption is that each of the matricisations can be approximated by low rank. This property can be assessed during the sampling, but of course the necessary rank can be arbitrary large, depending on the problem at hand. In the numerical examples we observe that the necessary rank varies, but for each concrete problem the sampling strategy appears to be quasi-optimal in the sense that the number of unknowns in a quasi-optimal approximation of the map in the hierarchical tensor format is proportional to the number of samples required to find a quasi-optimal approximation.

References

[1] I. Babŭska, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Num. Anal.*, 45:1005–1034, 2007.

[2] J. Bäck, F. Nobile, L. Tamellini, and R. Tempone. Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: A numerical comparison. In *Spectral and High Order Methods for Partial Differential Equations*, volume 76 of *Lecture Notes in Computational Science and Engineering*, pages 43–62. Springer, 2011.

[3] J. Ballani and L. Grasedyck. Tree adaptive approximation in the hierarchical tensor format. Preprint 141, DFG SPP-1324, 2013.

[4] J. Ballani, L. Grasedyck, and M. Kluge. Black box approximation of tensors in hierarchical Tucker format. *Linear Algebra Appl.*, 438(2):639–657, 2013.

[5] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.

[6] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[7] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numer.*, 7:1–49, 1998.

[8] P. Chen, A. Quarteroni, and G. Rozza. Comparison between reduced basis and stochastic collocation methods for elliptic problems. *J. Sci. Comput.*, published online DOI:10.1007/s10915-013-9764-2, 2013.

[9] A. Chkifa, A. Cohen, and C. Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Found. Comput. Math.*, published online DOI:10.1007/s10208-013-9154-z, 2013.

[10] J. Dick, F. Y. Kuo, Q. T. Le Gia, D. Nuyens, and C. Schwab. Higher order QMC Galerkin discretization for parametric operator equations. Preprint 2013-29, SAM ETH Zürich, 2013.

[11] M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies, and P. Wähnert. Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats. *Comput. Math. Appl.*, in press, 2012.

[12] 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*, volume 88 of *Lecture Notes in Computational Science and Engineering*, pages 31–56. Springer, 2013.

[13] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Lin. Alg. Appl.*, 261:1–22, 1997.

[14] L. Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.*, 31:2029–2054, 2010.

[15] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM, Math. Model. Numer. Anal.*, 41(3):575–605, 2007.

[16] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus.* Springer, Berlin, 2012.

[17] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *J. Fourier Anal. Appl.*, 15(5):706–722, 2009.

[18] B. N. Khoromskij and I. Oseledets. Quantics-TT collocation approximation of parameter-dependent and stochastic elliptic PDEs. *Comp. Meth. in Applied Math.*, 10(4):376–394, 2010.

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

[20] D. Kressner and C. Tobler. Low-rank tensor Krylov subspace methods for parameterized linear systems. *SIAM J. Matrix Anal. Appl.*, 32(4):1288–1316, 2011.

[21] F. Y. Kuo, C. Schwab, and I. H. Sloan. Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM J. Numer. Anal.*, 50(6):3351–3374, 2012.

[22] M. Loève. *Probability therory II, 4th edn.* Springer, New York, 1978.

[23] H. G. Matthies and E. Zander. Solving stochastic systems with low-rank tensor compression. *Linear Algebra Appl.*, 436(10):3819–3838, 2012.

[24] G. Migliorati, F. Nobile, E. von Schwerin, and R. Tempone. Approximation of quantities of interest in stochastic PDEs by the random discrete L^2 projection on polynomial spaces. *SIAM J. Sci. Comput.*, 35(3):A1440–A1446, 2013.

[25] N. C. Nguyen and J. Peraire. An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. *Int. J. Numer. Meth. Engng.*, 76:27–55, 2008.

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

[27] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.*, 15:229–275, 2008.

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

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

[30] C. Tobler. *Low-rank Tensor Methods for Linear Systems and Eigenvalue Problems.* PhD thesis, ETH Zürich, 2012.