

MATHICSE Technical Report

Nr. 27.2015
October 2015



Reduced basis methods: from low-rank matrices to low-rank tensors

Jonas Ballani, Daniel Kressner

Reduced basis methods: from low-rank matrices to low-rank tensors

Jonas Ballani* Daniel Kressner*

October 2, 2015

We propose a novel combination of the reduced basis method with low-rank tensor techniques for the efficient solution of parameter-dependent linear systems in the case of several parameters. This combination, called *rbTensor*, consists of three ingredients. First, the underlying parameter-dependent operator is approximated by an explicit affine representation in a low-rank tensor format. Second, a standard greedy strategy is used to construct a problem-dependent reduced basis. Third, the associated reduced parametric system is solved for *all* parameter values on a tensor grid simultaneously via a low-rank approach. This allows us to explicitly represent and store an approximate solution for all parameter values at a time. Once this approximation is available, the computation of output functionals and the evaluation of statistics of the solution becomes a cheap online task, without requiring the solution of a linear system.

1 Introduction

In this paper, we consider linear systems of the form

$$A(\mu)u(\mu) = f(\mu) \tag{1}$$

which depend on a parameter vector $\mu \in \mathcal{D} \subset \mathbb{R}^p$ with p not necessarily small. It is assumed that $A(\mu)$ is invertible on \mathcal{D} . Systems of the form (1) arise, e.g., in uncertainty quantification or optimization. In this context, the solution of (1) for many different parameter values $\mu \in \mathcal{D}$ is of interest, which calls for an efficient approximation of the solution map $\mu \mapsto u(\mu)$.

*MATHICSE-ANCHP, École Polytechnique Fédérale de Lausanne, Station 8, 1015 Lausanne, Switzerland, {jonas.ballani,daniel.kressner}@epfl.ch. The first author has been supported by an EPFL fellowship through the European Union's Seventh Framework Programme under grant agreement no. 291771.

In the described many-query setting, reduced basis methods provide a common framework to tackle a wide variety of parametric problems in a unified manner; we refer to the recent monographs [16, 30] for an introduction. Reduced basis methods rely on two major ingredients: First, the set of all solutions $\{u(\mu) : \mu \in \mathcal{D}\}$ needs to be well approximated by a low-dimensional subspace. Given a basis $V_N = [\xi_1, \dots, \xi_N]$ of such a subspace, this allows to approximate each solution $u(\mu)$ in terms of the expansion $u(\mu) \approx u_N(\mu) = \sum_{i=1}^N u_{N,i}(\mu)\xi_i$ with parameter-dependent coefficients $u_{N,i}$. The reduced basis V_N is typically constructed from so-called snapshots $u(\mu^i)$, $i = 1, \dots, N$, of the solution at carefully chosen sample points $\mu^i \in \mathcal{D}_{\text{train}}$ taken from a finite training set $\mathcal{D}_{\text{train}} \subset \mathcal{D}$.

Second, the operator A needs to admit, explicitly or implicitly, a separable representation of the form

$$A(\mu) = \sum_{q=1}^Q \theta_q(\mu)A_q \quad (2)$$

with parameter-dependent scalar functions θ_q and parameter-independent operators A_q . Similarly, the right-hand side f needs to admit a separable representation. Combined with the reduced basis, such representations constitute the basis for an efficient offline-online strategy, shifting the main computational cost to an expensive offline phase that is only carried out once. In this way, the cost in the online phase for solving one linear system (1) is reduced to the assembly and solution of a (much smaller) N -dimensional linear system. Note that the separability assumption is essential. If this assumption is violated, it is sometimes possible to replace A and f by (good) separable approximations. Such an approximation may, e.g., be obtained by the empirical interpolation method and its variants [5, 13, 17, 28].

The reduced basis method can be considered as a low-rank method in the sense that it exploits the separability of the operator A , the right-hand side f , and the solution u into a spatially-dependent component and a parameter-dependent part. Often, there is additional low-rank structure inside the parameter-dependent part. For example, when considering (2), the functions $\theta_q(\mu)$ may admit a separable representation of the form $\theta_q(\mu) = \sum_{j=1}^k \theta_q^{j,1}(\mu_1)\theta_q^{j,2}(\mu_2) \cdots \theta_q^{j,p}(\mu_p)$. In particular, this is the case, with $k = 1$, for an operator given in the affine form

$$A(\mu) = A_0 + \mu_1 A_1 + \dots + \mu_p A_p. \quad (3)$$

It is our goal to leverage this additional structure for an explicit approximation of the solution u on the entire parameter domain \mathcal{D} , which does not only allow for an efficient online evaluation phase, like in standard reduced basis methods, but is also appropriate for the direct computation of statistics and for optimization tasks.

In our framework, we assume that the parameter domain $\mathcal{D} \subset \mathbb{R}^p$ is a p -dimensional box, which is discretized by a tensor product grid $\mathcal{D}_{\mathcal{I}} \subset \mathcal{D}$. The set of all solutions on $\mathcal{D}_{\mathcal{I}}$ can thus be arranged as a tensor of order $p + 1$, which satisfies a (huge) global linear system. This suggests the use of low-rank tensor techniques; see [14, 12] for an overview. Existing techniques for solving the global linear system arising from (1)

include standard iterative schemes combined with low-rank truncation [1, 8, 21, 22, 25, 26, 33], local optimization on low-rank representations [9, 10, 19, 24, 29], and sampling techniques [2, 27]. In contrast to these approaches, we solve the global linear system only in the reduced space that is constructed incrementally in the course of the reduced basis method. In particular, this has the advantage that the size of the linear system becomes independent of the spatial discretization.

The method proposed in this work, *rbTensor*, contains the following ingredients:

- (1) an adaptive approximation of the operator-valued function $A(\mu)$ on $\mathcal{D}_{\mathcal{I}}$ by a low-rank tensor representation;
- (2) a greedy strategy for constructing a problem-dependent reduced basis with a training set $\mathcal{D}_{\text{train}} \subset \mathcal{D}_{\mathcal{I}}$;
- (3) an adaptive solution of the reduced global linear system.

Here, adaptivity refers to the automatic detection of the required low-rank representation to attain a given accuracy. As we only require queries to the operator A , the right-hand side f and the solution u for specific parameter values $\mu \in \mathcal{D}$, we expect to be able to deal with problems of a rather general nature.

The rest of the paper is organized as follows. In Section 2 we shortly review the main ingredients of the reduced basis method. We then formulate the parametric problem on a tensor grid in Section 3 and recall the hierarchical tensor format from [15]. Section 4 is dedicated to the low-rank approximation of the operator A by means of a combination of the matrix discrete empirical interpolation method (MDEIM) from [28] and the cross approximation technique from [3]. Afterwards, we discuss the solution of the reduced global linear system in Section 5. In the numerical experiments in Section 6, we illustrate the potential as well as limitations of our approach by three examples from the literature.

2 The Reduced Basis Method

In this section, we briefly recall the reduced basis method for solving parametrized partial differential equations (PDEs), following the standard setting described in the literature [16, 30].

Variational formulation and discretization. Let X be a Hilbert space, $a(\cdot, \cdot; \mu) : X \times X \rightarrow \mathbb{R}$ a continuous, symmetric and coercive bilinear form for all $\mu \in \mathcal{D} \subset \mathbb{R}^p$ and let $f(\cdot; \mu) \in X'$. We seek $u(\mu) \in X$ such that for all $\mu \in \mathcal{D}$

$$a(u, v; \mu) = f(v; \mu), \quad \forall v \in X. \quad (4)$$

For a *truth* space $X^{\mathcal{N}} \subset X$ of dimension $\mathcal{N} \in \mathbb{N}$, we approximate (4) by seeking a *truth* solution $u^{\mathcal{N}}(\mu) \in X^{\mathcal{N}}$ such that for all $\mu \in \mathcal{D}$

$$a(u^{\mathcal{N}}, v; \mu) = f(v; \mu), \quad \forall v \in X^{\mathcal{N}}. \quad (5)$$

Given a particular basis $\{\varphi_1, \dots, \varphi_N\}$ of X^N , we expand the solution $u^N(\mu)$ as

$$u^N(\mu) = \sum_{i=1}^N u_i^N(\mu) \varphi_i.$$

Problem (5) is then equivalent to the linear system

$$A(\mu)u^N(\mu) = f^N(\mu) \quad (6)$$

with the matrix $A \in \mathbb{R}^{N \times N}$ defined by $A_{i,j} = a(\varphi_j, \varphi_i; \mu)$, the right-hand side $f^N \in \mathbb{R}^N$ given by $f^N = [f(\varphi_1; \mu), \dots, f(\varphi_N; \mu)]^\top$, and the solution vector $u^N = [u_1^N, \dots, u_N^N]^\top$. By our assumptions, $A(\mu)$ is symmetric and positive definite for all $\mu \in \mathcal{D}$.

Construction of reduced basis. From the solution u^N of (6) at N different parameter values μ^1, \dots, μ^N , we construct a subspace $X_N \subset \mathbb{R}^N$ by setting

$$X_N := \text{span}\{u^N(\mu^i) : i = 1, \dots, N\}.$$

We assume that X_N has dimension N . These spaces are nested in the sense that $X_1 \subset X_2 \subset \dots \subset X_N$.

For an arbitrary parameter $\mu \in \mathcal{D}$, the approximate solution $u_N^N(\mu) \in X_N$ is determined by the Galerkin condition

$$A(\mu)u_N^N(\mu) - f^N(\mu) \perp X_N. \quad (7)$$

Given an (orthonormal) basis $V_N \in \mathbb{R}^{N \times N}$ of X_N , we can write $u_N^N(\mu) = V_N u_N(\mu)$ for some vector $u_N(\mu) \in \mathbb{R}^N$. The condition (7) is then equivalent to the linear system

$$A_N(\mu)u_N(\mu) = f_N(\mu) \quad (8)$$

with the reduced matrix $A_N(\mu) = V_N^\top A(\mu) V_N \in \mathbb{R}^{N \times N}$ and the reduced right-hand side $f_N(\mu) = V_N^\top f^N(\mu) \in \mathbb{R}^N$.

Affine Parameter Dependence. As a key ingredient, the reduced basis method relies on the separability of the parameters from the operator, see (2). In particular, we have

$$A(\mu) = \sum_{q=1}^Q \theta_q(\mu) A_q \quad (9)$$

for parameter-independent matrices A_q . As a consequence, the reduced matrix A_N in (8) satisfies

$$A_N(\mu) = \sum_{q=1}^Q \theta_q(\mu) A_{N,q}, \quad A_{N,q} := V_N^\top A_q V_N \in \mathbb{R}^{N \times N}.$$

The reduced matrices $A_{N,q}$ are independent of μ and thus only need to be assembled once. Combined with an analogous reasoning for a right-hand side with affine parameter dependence, this forms the basis for an efficient offline-online strategy.

Greedy selection of snapshots. The parameters $\mu^1, \dots, \mu^N \in \mathcal{D}$ for the construction of the reduced basis spaces X_1, \dots, X_N are typically chosen in a greedy manner. As an essential ingredient, one needs a reasonably tight error bound $\Delta_N : \mathcal{D} \rightarrow \mathbb{R}$ for which

$$\|\mathbf{u}^{\mathcal{N}}(\mu) - \mathbf{u}_N^{\mathcal{N}}(\mu)\| \leq \Delta_N(\mu), \quad (10)$$

with a suitably chosen norm $\|\cdot\|$. For problems arising from elliptic partial differential equations, such an error bound can be constructed from the residual and a lower bound on the coercivity constant; see [31] for examples.

However, it is usually impossible to control the error (10) a priori on the entire (continuous) set \mathcal{D} . To deal with this issue, it is common practice to replace \mathcal{D} by a finite surrogate training set $\mathcal{D}_{\text{train}} \subset \mathcal{D}$. Algorithm 1 summarizes the resulting greedy approach, which terminates when (10) is below a user-specified tolerance ε_{RB} for all $\mu \in \mathcal{D}_{\text{train}}$.

Algorithm 1 Greedy Reduced Basis Construction

```

1:  $N := 0, V_0 := []$ 
2: while  $\max_{\mu \in \mathcal{D}_{\text{train}}} \Delta_N(\mu) > \varepsilon_{\text{RB}}$  do
3:    $N := N + 1$ 
4:    $\mu^N := \arg \max_{\mu \in \mathcal{D}_{\text{train}}} \Delta_{N-1}(\mu)$ 
5:   Solve  $A(\mu^N) \mathbf{u}^{\mathcal{N}}(\mu^N) = \mathbf{f}^{\mathcal{N}}(\mu^N)$ 
6:    $V_N := \text{orth}[V_{N-1}, \mathbf{u}^{\mathcal{N}}(\mu^N)]$ 
7: end while

```

Online evaluation phase. Algorithm 1, together with the construction of the error bound $\Delta_N(\mu)$, constitutes the offline phase of a typical reduced basis method. In the online phase, a new query to the solution vector $\mathbf{u}^{\mathcal{N}}(\mu)$ is processed by assembling the $N \times N$ linear system (8), solving this reduced linear system, and returning the approximation $V_N \mathbf{u}_N(\mu)$. The computational complexity for these three tasks is $\mathcal{O}(QN^2)$, $\mathcal{O}(N^3)$, and $\mathcal{O}(NN)$, respectively. The bound (10) provides an a posteriori estimate of the error.

3 Tensor-structured Formulation

In this section, we show how the reduced basis method discussed above can benefit from additional structure in the parameter domain, by means of low-rank tensor approximation.

3.1 Discrete Formulation

In what follows, we assume that the parameter domain $\mathcal{D} \subset \mathbb{R}^p$ has Cartesian product structure $\mathcal{D} = \mathcal{D}_1 \times \dots \times \mathcal{D}_p$. We discretize each $\mathcal{D}_i \subset \mathbb{R}$ with a grid $\mathcal{D}_{\mathcal{I}_i} := \{\mu_{i,1}, \dots, \mu_{i,n_i}\} \subset \mathcal{D}_i$ with n_i points indexed by $\mathcal{I}_i := \{1, \dots, n_i\}$. This defines a tensor grid

$$\mathcal{D}_{\mathcal{I}} := \mathcal{D}_{\mathcal{I}_1} \times \dots \times \mathcal{D}_{\mathcal{I}_p} \subset \mathcal{D} \subset \mathbb{R}^p$$

with $\mathcal{I} := \mathcal{I}_1 \times \dots \times \mathcal{I}_p$. After restricting \mathcal{D} to $\mathcal{D}_{\mathcal{I}}$, our goal becomes the solution of the linear system

$$A(\mu)u^{\mathcal{N}}(\mu) = f^{\mathcal{N}}(\mu) \quad \forall \mu \in \mathcal{D}_{\mathcal{I}}. \quad (11)$$

These $n_{\mathcal{I}} := \#\mathcal{I}$ linear systems can be arranged in one global, block diagonal linear system of size $n_{\mathcal{I}} \cdot \mathcal{N}$:

$$\mathcal{A}x = b, \quad (12)$$

where

$$\mathcal{A} := \begin{pmatrix} A(\mu^1) & & \\ & \ddots & \\ & & A(\mu^{n_{\mathcal{I}}}) \end{pmatrix}, \quad x := \begin{pmatrix} u^{\mathcal{N}}(\mu^1) \\ \vdots \\ u^{\mathcal{N}}(\mu^{n_{\mathcal{I}}}) \end{pmatrix}, \quad b := \begin{pmatrix} f^{\mathcal{N}}(\mu^1) \\ \vdots \\ f^{\mathcal{N}}(\mu^{n_{\mathcal{I}}}) \end{pmatrix}.$$

Note that \mathcal{A} inherits the symmetry and positive definiteness from $A(\mu)$.

Analogously, the corresponding reduced linear systems (8),

$$A_N(\mu)u_N(\mu) = f_N(\mu) \quad \forall \mu \in \mathcal{D}_{\mathcal{I}}, \quad (13)$$

can be arranged into a global linear system of size $n_{\mathcal{I}} \cdot N$:

$$\mathcal{A}_N x_N = b_N, \quad (14)$$

where

$$\mathcal{A}_N := (I_{\mathcal{I}} \otimes V_N^{\top}) \mathcal{A} (I_{\mathcal{I}} \otimes V_N), \quad x_N := (u_N^{\top}(\mu^1), \dots, u_N^{\top}(\mu^{n_{\mathcal{I}}}))^{\top}, \quad b_N := (I_{\mathcal{I}} \otimes V_N^{\top}) b,$$

and $I_{\mathcal{I}} \in \mathbb{R}^{n_{\mathcal{I}} \times n_{\mathcal{I}}}$ is the identity matrix.

Because of their size, the formulations (12) and (14) are of no use unless the involved quantities \mathcal{A}, x, b , and \mathcal{A}_N, x_N, b_N can be represented in an efficient way. To this end, we will use tensor approximation techniques.

3.2 Hierarchical Tensor Format

Considering the global linear system (12), the tensor structure of \mathcal{I} allows us to view the vectors $x, b \in \mathbb{R}^{n_{\mathcal{I}} \cdot \mathcal{N}}$ as tensors \mathcal{X}, \mathcal{B} of order $p+1$ and size $n_1 \times n_2 \times \dots \times n_p \times \mathcal{N}$. Similarly, the vectors $x_N, b_N \in \mathbb{R}^{n_{\mathcal{I}} \cdot N}$ in the global reduced linear system (14) can be viewed as tensors $\mathcal{X}_N, \mathcal{B}_N$ of size $n_1 \times n_2 \times \dots \times n_p \times N$. In the following, we recall the hierarchical tensor format introduced in [15] and analyzed in [11] for approximating these tensors.

Let $\mathcal{Y} \in \mathbb{R}^{\mathcal{J}}$ be a tensor of order $d \in \mathbb{N}$ with its entries indexed by a finite product index set $\mathcal{J} = \mathcal{J}_1 \times \dots \times \mathcal{J}_d$. We start by defining a broad class of matricizations of \mathcal{Y} .

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

$$\mathcal{M}_t : \mathbb{R}^{\mathcal{J}} \rightarrow \mathbb{R}^{\mathcal{J}_t} \otimes \mathbb{R}^{\mathcal{J}_{[t]}}, \quad \mathcal{J}_t := \times_{i \in t} \mathcal{J}_i, \quad \mathcal{J}_{[t]} := \times_{i \in [t]} \mathcal{J}_i,$$

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

$$\mathcal{M}_t(\mathcal{Y})_{(j_i)_{i \in t}, (j_i)_{i \in [t]}} := \mathcal{Y}_{(j_1, \dots, j_d)}, \quad j = (j_1, \dots, j_d) \in \mathcal{J}. \quad (15)$$

Note that we always assume that the lexicographical order is used for arranging the multi-indices into the row and columns of the matrix $\mathcal{M}_t(\mathcal{Y})$ in (15).

In order to allow for an efficient hierarchical representation, the subsets $t \subset D$ are organized in a binary tree T_D with root D such that each node $t \in T_D$ is non-empty and each $t \in T_D$ with $\#t \geq 2$ is the disjoint union of its sons $t_1, t_2 \in T_D$, cf. Figure 1.

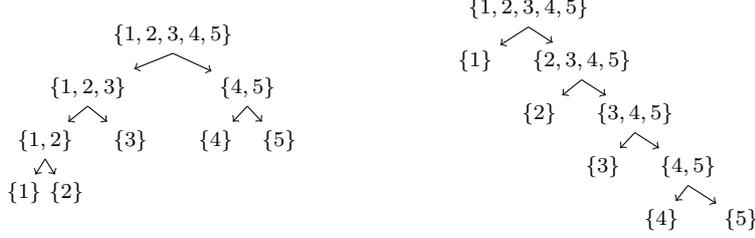


Figure 1: Dimension trees T_D for $d = 5$. Left: Balanced tree. Right: Linear tree.

Based on the concept of the matricization of tensors and the definition of a dimension tree, the hierarchical tensor format is defined as follows.

Definition 2 (hierarchical rank, hierarchical format). Let T_D be a dimension tree. The *hierarchical rank* $\mathbf{k} := (k_t)_{t \in T_D}$ of a tensor $\mathcal{Y} \in \mathbb{R}^{\mathcal{J}}$ is defined by

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

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

$$\mathcal{H}_{\mathbf{k}} := \{\mathcal{Y} \in \mathbb{R}^{\mathcal{J}} : \text{rank}(\mathcal{M}_t(\mathcal{Y})) \leq k_t, t \in T_D\}.$$

Given a tensor $\mathcal{Y} \in \mathcal{H}_{\mathbf{k}}$, Definition 2 implies that there are matrices $U_t \in \mathbb{R}^{\mathcal{J}_t \times k_t}$, with $\mathcal{J}_t := \times_{i \in t} \mathcal{J}_i$, such that the column range of $\mathcal{M}_t(\mathcal{Y})$ is contained in the column range of U_t for every $t \in T_D$. One usually chooses U_D to represent \mathcal{Y} and all other U_t to have orthonormal columns. The following lemma is key to representing such matrices U_t and, in turn, the tensor \mathcal{Y} compactly. Note that we use $(A)_{\cdot, \ell}$ to denote the ℓ th column of a matrix A .

Lemma 3 (cf. [11]). *Let $\mathcal{Y} \in \mathcal{H}_{\mathbf{k}}$ and consider matrices $U_t \in \mathbb{R}^{\mathcal{J}_t \times k_t}$ satisfying the properties described above. For every (non-leaf) node $t \in T_D$ with sons $t_1, t_2 \in T_D$, there is a tensor $B_t \in \mathbb{R}^{k_t \times k_{t_1} \times k_{t_2}}$ such that*

$$(U_t)_{\cdot, \ell} = \sum_{\ell_1=1}^{k_{t_1}} \sum_{\ell_2=1}^{k_{t_2}} (B_t)_{\ell, \ell_1, \ell_2} (U_{t_1})_{\cdot, \ell_1} \otimes (U_{t_2})_{\cdot, \ell_2}, \quad \ell = 1, \dots, k_t. \quad (16)$$

Thanks to the recursion (16), one only needs to store the so called *transfer tensors* B_t in each inner node $t \in T_D$ and the matrices U_t for all leaf nodes $t \in T_D$. The complexity for the hierarchical representation sums up to $\mathcal{O}(dk^3 + dkn)$, where $k := k_{\max} = \max_{t \in T_D} k_t$, $n := \max_{i \in D} \#\mathcal{J}_i$. The *effective rank* k_{eff} is the real positive number such that $(d-1)k_{\text{eff}}^3 + dk_{\text{eff}}n$ is the actual storage cost for a tensor in $\mathcal{H}_{\mathbf{k}}$. This quantity is particularly of interest when the ranks k_t differ significantly for some nodes $t \in T_D$.

4 Low-rank Operator Approximation

In order to use low-rank tensor formats for solving linear systems, we also need to (approximately) represent the involved linear operator in a way that conforms with the format. In some cases, this can be trivially achieved, for example, when the parameter dependence is in the affine form (3). To deal with arbitrary nonlinear parameter dependence, we will develop a blackbox procedure for achieving such a representation.

Given a general matrix $A(\mu) \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$, which is *not* necessarily in the form (9), the aim of this section is to construct an explicit separable approximation of the form

$$A(\mu^i) \approx \sum_{q=1}^Q \Theta_{(i,q)} A_q, \quad \mu^i \in \mathcal{D}_{\mathcal{I}}, \quad (17)$$

with parameter-independent matrices $A_q \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ and a tensor $\Theta \in \mathbb{R}^{n_{\mathcal{I}} \times Q}$ that can be represented in the hierarchical tensor format. To this end, we perform the following two steps:

Step 1. find an orthogonal matrix $W \in \mathbb{R}^{\mathcal{N}^2 \times Q}$ such that $\text{span}\{\text{vec}(A(\mu)) : \mu \in \mathcal{D}_{\mathcal{I}}\} \approx \text{range}(W)$,

Step 2. construct $\Theta \in \mathcal{H}_{\mathbf{k}}$ such that $\Theta_{(i,\cdot)} \approx W^\top \text{vec}(A(\mu^i))$ for all $\mu^i \in \mathcal{D}_{\mathcal{I}}$.

From the columns of W , we obtain the parameter-independent matrices A_q by reversing the vectorization operation, i.e.

$$A_q = \text{unvec}(W_{\cdot,q}), \quad q = 1, \dots, Q. \quad (18)$$

Note that we do not require the matrices A_q to be realizations of the operator $A(\mu)$ for some $\mu \in \mathcal{D}_{\mathcal{I}}$ as this does not comply with the orthogonality requirement on W .

4.1 Review of MDEIM

Step 1 of the operator approximation goes along the lines of the so-called Matrix Discrete Empirical Interpolation Method (MDEIM); see [28] and the references therein. For any $\mu \in \mathcal{D}_{\mathcal{I}}$, we interpret $A(\mu) \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ as a vector $\text{vec}(A) \in \mathbb{R}^{\mathcal{N}^2}$. For sparse matrices $A(\mu)$, we actually neglect entries that are zero for all $\mu \in \mathcal{D}_{\mathcal{I}}$ and only consider the non-zero entries.

In Step 1, we construct a matrix

$$W = \text{orth}[\text{vec}(A(\mu^1)), \dots, \text{vec}(A(\mu^Q))],$$

where μ^1, \dots, μ^Q are suitably selected parameters from a training set $\mathcal{D}_{\text{op}} \subset \mathcal{D}_{\mathcal{I}}$. In view of (17), we aim at attaining a good approximation

$$\text{vec}(A(\mu^i)) \approx WW^\top \text{vec}(A(\mu^i)), \quad \mu^i \in \mathcal{D}_{\mathcal{I}}. \quad (19)$$

Using a greedy strategy for selecting these parameters leads to Algorithm 2.

Algorithm 2 Matrix Discrete Empirical Interpolation Method

- 1: $W := []$
 - 2: **repeat**
 - 3: $\mu^* := \arg \max_{\mu \in \mathcal{D}_{\text{op}}} \|(I - WW^\top) \text{vec}(A(\mu))\|_2$
 - 4: $W := \text{orth}[W, \text{vec}(A(\mu^*))]$
 - 5: **until** $\|(I - WW^\top) \text{vec}(A(\mu^*))\|_2 \leq \varepsilon_{\text{op}}$
-

Implementation aspects of Algorithm 2:

- The orthogonalization step in line 4 is performed by standard Gram-Schmidt orthogonalization applied to the new basis vector, possibly followed by reorthogonalization.
- The training set \mathcal{D}_{op} serves as a surrogate for $\mathcal{D}_{\mathcal{I}}$ in our aim to determine an accurate operator approximation for *all* parameters $\mu \in \mathcal{D}_{\mathcal{I}}$. It is chosen significantly smaller than $\mathcal{D}_{\mathcal{I}}$ in order to keep the computational effort spent on Algorithm 2 under control. In all numerical experiments we use the following strategy for constructing \mathcal{D}_{op} inspired by tensor cross approximation [12, Sec. 3.5]. First, we randomly determine some pivot index $i \in \mathcal{I}$. We then consider the set

$$\mathcal{I}_{\text{cross}}(i) := \{(i_1, \dots, i_{\ell-1}, j, i_{\ell+1}, \dots, i_p) : j \in \mathcal{I}_\ell, \ell = 1, \dots, p\} \subset \mathcal{I} \quad (20)$$

which forms a 'cross' with center i . Repeating this strategy a few number s of times (say $s = 3$) for random indices $i^1, \dots, i^s \in \mathcal{I}$, we arrive at

$$\mathcal{I}_{\text{op}} := \mathcal{I}_{\text{cross}}(i^1) \cup \dots \cup \mathcal{I}_{\text{cross}}(i^s),$$

which determines the training set for the first loop of Algorithm 2:

$$\mathcal{D}_{\text{op}} := \{\mu_i \in \mathcal{D}_{\mathcal{I}} : i \in \mathcal{I}_{\text{op}}\}.$$

In every subsequent loop of Algorithm 2, this set is enriched with s additional (random) crosses. In line 3, we reuse the information computed in the previous loops of the algorithm as much as possible.

- The parameter ε_{op} used in the stopping criterion of line 5 needs to be carefully chosen. It directly influences the attainable accuracy for the solution of the linear system (11). Standard perturbation results [18, Thm. 2.7] indicate that ε_{op} may get amplified by the norm of the inverse of the operator; see Section 6, in particular Figure 7.

4.2 Tensor Approximation

Once an orthogonal matrix $W \in \mathbb{R}^{\mathcal{N}^2 \times Q}$ satisfying (19) is constructed, we have obtained a separable approximation of the form

$$A(\mu^i) \approx \sum_{q=1}^Q \Theta_{(i,q)} A_q, \quad \Theta_{(i,\cdot)} = W^\top \text{vec}(A(\mu^i)), \quad \mu^i \in \mathcal{D}_{\mathcal{I}},$$

with A_q as in (18). In Step 2 of our low-rank operator approximation, we consider $\Theta \in \mathbb{R}^{n_{\mathcal{I}} \cdot Q}$ as a tensor $\Theta \in \mathbb{R}^{\mathcal{J}}$ of order $p+1$ over the index set $\mathcal{J} = \mathcal{I}_1 \times \cdots \times \mathcal{I}_p \times \{1, \dots, Q\}$, and replace it by a low-rank approximation in the hierarchical tensor format $\mathcal{H}_{\mathbf{k}}$. In [3], a general adaptive strategy for the black box approximation of tensors in $\mathcal{H}_{\mathbf{k}}$ has been proposed, which we apply to this particular setting.

Let $D = \{1, \dots, p+1\}$ and let T_D be a dimension tree as introduced in Section 3.2. The main idea of the approach in [3] is to recursively approximate the matricizations of $M = \mathcal{M}_t(\Theta)$ at any node $t \in T_D$ by a so-called *cross approximation* of the form

$$M \approx \tilde{M} := M|_{\mathcal{J}_t \times \mathcal{Q}_t} \cdot M|_{\mathcal{P}_t \times \mathcal{Q}_t}^{-1} \cdot M|_{\mathcal{P}_t \times \mathcal{J}_{[t]}} \quad (21)$$

with $\text{rank}(\tilde{M}) = k_t$ and pivot sets $\mathcal{P}_t \subset \mathcal{J}_t$, $\mathcal{Q}_t \subset \mathcal{J}_{[t]}$ of size k_t . For each node $t \in T_D$, the rank k_t can be chosen adaptively in order to reach a given (heuristic) target accuracy $\varepsilon_{\text{ten}} \geq 0$ s.t. $\|M - \tilde{M}\|_2 \approx \varepsilon_{\text{ten}} \|M\|_2$.

The matrices $M|_{\mathcal{J}_t \times \mathcal{Q}_t}$, $M|_{\mathcal{P}_t \times \mathcal{J}_{[t]}}$ in (21) are never formed explicitly. The essential information for the construction of $\Theta \in \mathcal{H}_{\mathbf{k}}$ with $\mathbf{k} = (k_t)_{t \in T_D}$ are condensed in the pivot sets $\mathcal{P}_t, \mathcal{Q}_t$ and the matrices $M|_{\mathcal{P}_t \times \mathcal{Q}_t} \in \mathbb{R}^{k_t \times k_t}$ from (21). This construction is explicit in the sense that the necessary transfer tensors B_t for all inner nodes $t \in T_D$ and the matrices U_t in the leaf nodes $t \in T_D$ are directly determined by the values of Θ at certain entries defined by the pivots sets. The details of this procedure can be found in [2, 3, 14].

Since each entry of Θ corresponds to the assembly of $A(\mu)$ for some $\mu \in \mathcal{D}_{\mathcal{I}}$ and a multiplication with W^\top , we need to keep the number of requested entries small when choosing the pivot sets $\mathcal{P}_t, \mathcal{Q}_t$ in each node $t \in T_D$. To this end, we use again training sets of the form (20). Assuming a balanced tree T_D , this strategy requires the evaluation of $\mathcal{O}(pk^3 + p \log(p)k^2n)$ entries with $n := \max_{i \in D} \#\mathcal{J}_i$, $k := \max_{t \in T_D} k_t$, see [3].

We orthogonalize and perform low-rank truncation, at the level of ε_{op} , of the tensor $\Theta \in \mathcal{H}_{\mathbf{k}}$ obtained from the cross approximation procedure described above. This often results in a slight reduction of the hierarchical rank \mathbf{k} .

Remark 4. Analogous to the low-rank approximation of the operator $A(\mu)$, we obtain an approximation of the right-hand side $\mathbf{f}^{\mathcal{N}}(\mu)$ from (11) by suitable modifications of Algorithm 2 and the tensor cross approximation technique discussed above.

5 Solution of the Linear System

Given low-rank approximations of the operator $A(\mu)$ and the right-hand side $\mathbf{f}^{\mathcal{N}}(\mu)$, our aim is now to find an approximate solution to (11) by means of low-rank tensor

techniques. Given a separable approximation of the form (17), we replace the linear operator \mathcal{A} in (12) by

$$\mathcal{A} = \sum_{q=1}^Q D_q \otimes A_q \quad (22)$$

with diagonal matrices $D_q := \text{diag}\{\Theta_{(\cdot,q)}\} \in \mathbb{R}^{n_{\mathcal{I}} \times n_{\mathcal{I}}}$ and $\Theta \in \mathcal{H}_{\mathbf{k}}$. Equivalently, \mathcal{A} can be viewed as a linear operator on the space of tensors $\mathbb{R}^{\mathcal{J}}$ with $\mathcal{J} = \mathcal{I}_1 \times \cdots \times \mathcal{I}_p \times \{1, \dots, \mathcal{N}\}$. To characterize the low-rank properties of the operator \mathcal{A} , we first state the following general lemma.

Lemma 5. *Let $\mathcal{J} = \mathcal{J}_1 \times \cdots \times \mathcal{J}_d$ and let T_D be a dimension tree. Consider a linear operator $\mathcal{A} : \mathbb{R}^{\mathcal{J}} \rightarrow \mathbb{R}^{\mathcal{J}}$ of the form*

$$\mathcal{A} = \sum_{j_1 \in \mathcal{K}_1} \cdots \sum_{j_d \in \mathcal{K}_d} \mathcal{C}_{(j_1, \dots, j_d)} (A_{j_1}^{(1)} \otimes \cdots \otimes A_{j_d}^{(d)}), \quad A_{j_i}^{(i)} \in \mathbb{R}^{\mathcal{J}_i \times \mathcal{J}_i},$$

with a tensor $\mathcal{C} \in \mathbb{R}^{\mathcal{K}}$ with $\mathcal{K} := \mathcal{K}_1 \times \cdots \times \mathcal{K}_d$ and let $\mathcal{X} \in \mathbb{R}^{\mathcal{J}}$. If $\mathcal{C} \in \mathcal{H}_{\mathbf{k}}$ and $\mathcal{X} \in \mathcal{H}_{\mathbf{r}}$ with respect to the tree T_D , then for $\mathcal{Y} := \mathcal{A}\mathcal{X}$ it holds that $\mathcal{Y} \in \mathcal{H}_{\tilde{\mathbf{r}}}$ where $\tilde{r}_t = k_t \cdot r_t$ for all $t \in T_D$.

Proof. For a proof, we refer to [14, Section 13.9]. □

In order to apply Lemma 5 to the operator \mathcal{A} from (22), note that we can trivially write

$$D_q = \sum_{j \in \mathcal{I}} \Theta_{(j,q)} (E_{j_1}^{(1)} \otimes \cdots \otimes E_{j_p}^{(p)}), \quad E_{j_i}^{(i)} := e_{j_i}^{(i)} e_{j_i}^{(i)\top},$$

where $e_{j_i}^{(i)} \in \mathbb{R}^{\mathcal{I}_i}$ is the j_i -th unit vector. With this preparation, the operator \mathcal{A} reads

$$\mathcal{A} = \sum_{j \in \mathcal{I}} \sum_{q=1}^Q \Theta_{(j,q)} (E_{j_1}^{(1)} \otimes \cdots \otimes E_{j_p}^{(p)} \otimes A_q)$$

which allows us to apply Lemma 5 with $\mathcal{K} = \mathcal{I}_1 \times \cdots \times \mathcal{I}_p \times \{1, \dots, Q\}$.

We conclude that an operator \mathcal{A} of the form (22) can be represented in a tensor-structured way, which allows for the use of low-rank tensor methods for solving linear systems

$$\mathcal{A}\mathcal{X} = \mathcal{B}, \quad (23)$$

provided that the tensor \mathcal{B} is in $\mathcal{H}_{\mathbf{k}}$. As a standard approach from the literature, we first consider a preconditioned iterative method combined with a rank truncation procedure, see [25] for examples.

5.1 Preconditioned iterative solution with truncation

Originating from the discretization of a parameterized PDE, the symmetric positive definite operator \mathcal{A} can be expected to become ill-conditioned and therefore an iterative method applied to (23) needs to be combined with an efficient preconditioner. We consider rank-one preconditioners of the form

$$\mathcal{P} := I_{\mathcal{I}} \otimes P,$$

with $P \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ defined by

$$P := \sum_{q=1}^Q \lambda_q A_q,$$

with suitably chosen weights λ_q , for example, $\lambda_q = \theta_q(\bar{\mu})$ for some $\bar{\mu} \in \mathcal{D}$. A preconditioner of this simple form mainly addresses the dependence of the convergence of the iterative method on the dimension \mathcal{N} of the truth finite element space.

We apply a truncated CG method, using the preconditioner described above, with a truncation strategy similar to [23]. From each iterate \mathcal{X}^i in the CG method, we need to compute the residual $\mathcal{R}^i = \mathcal{B} - \mathcal{A}\mathcal{X}^i$ and the conjugate directions. To keep the tensor ranks small, we truncate the iterates and the corresponding residuals in each step of the CG method. As soon as an iterate \mathcal{X}^i approaches the solution \mathcal{X} , the residual \mathcal{R}^i will become small such that there is no need to use an exaggerated tolerance close to the solution. We therefore approximate \mathcal{X}^i by $\tilde{\mathcal{X}}^i$ such that $\|\mathcal{X}^i - \tilde{\mathcal{X}}^i\|_2 \leq \varepsilon \|\mathcal{X}^i\|_2$ whereas the residual is truncated such that $\|\mathcal{R}^i - \tilde{\mathcal{R}}^i\|_2 \leq \varepsilon_i \|\mathcal{R}^i\|_2$ with $\varepsilon_i := C\varepsilon \|\mathcal{R}^0\|_2 / \|\mathcal{R}^{i-1}\|_2$ and $C := 0.1$.

To illustrate the performance of the truncated CG method, we consider the linear system (12) arising from the cookie problem in parameter dimension $p = 9$ described in more detail in Section 6.1. Figure 2 clearly shows that the attainable relative residual depends on the prescribed tolerance ε . In each iteration, we monitor the ranks of the iterates and the residual. While the ranks of the iterates \mathcal{X}^i remain moderate, the ranks of the residual \mathcal{R}^i quickly grow in the first few iterations and get smaller as soon as the attainable accuracy is reached. The intermediate large ranks in the residual have a strong negative impact on efficiency; for example, the complexity of tensor truncation is $\mathcal{O}(k^4)$.

5.2 Successive solution of the reduced linear system

Instead of solving the large tensor system (23) all at once, we propose to approximate the solution \mathcal{X} by successively solving the reduced problems (14) for $N = 1, 2, \dots, N_{\max}$. Our main strategy is given in Algorithm 3. Line 7 corresponds to the solution of the reduced system for *all* parameter values $\mu \in \mathcal{D}_{\mathcal{I}}$ and is hence the major difference to the standard reduced basis construction from Algorithm 1.

Analogous to the previous section, the reduced system in line 7 could be solved iteratively. To accelerate the convergence, we could again use a rank-one preconditioner

$$\mathcal{P}_N := I_{\mathcal{I}} \otimes P_N$$

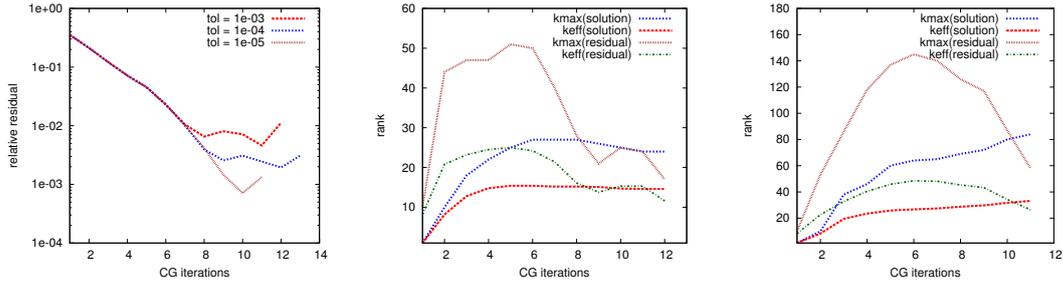


Figure 2: Truncated CG method for the cookie problem with $p = 9$ parameters: Left: relative residual depending on the tolerance ε . Effective and maximal ranks of the iterates and the residual in the hierarchical tensor format for truncation parameter $\varepsilon = 10^{-3}$ (center) and for $\varepsilon = 10^{-5}$ (right).

Algorithm 3 Greedy RB Low-rank Tensor Approximation

- 1: $N := 0, V_0 := []$
 - 2: **while** $\max_{\mu \in \mathcal{D}_{\text{train}}} \Delta_N(\mu) > \varepsilon_{\text{RB}}$ **do**
 - 3: $N := N + 1$
 - 4: $\mu^N := \arg \max_{\mu \in \mathcal{D}_{\text{train}}} \Delta_{N-1}(\mu)$
 - 5: Solve $A(\mu^N) \mathbf{u}^N(\mu^N) = \mathbf{f}^N(\mu^N)$
 - 6: $V_N := \text{orth}[V_{N-1}, \mathbf{u}^N(\mu^N)]$
 - 7: Solve $\mathcal{A}_N \mathcal{X}_N = \mathcal{B}_N$ from (8) in $\mathcal{H}_{\mathbf{k}}$
 - 8: **end while**
-

with $P_N \in \mathbb{R}^{N \times N}$ defined by

$$P_N := \sum_{q=1}^Q \lambda_q A_{N,q}, \quad A_{N,q} = V_N^\top A_q V_N.$$

Since N can be expected to remain small, the matrix P_N can be explicitly inverted. However, for increasing N we can expect the same problematic rank behavior observed in Figure 2, because the reduced spaces X_N tend to represent increasingly better approximations of the truth finite element space X^N . Our numerical experiments confirm this expectation.

To circumvent the problems inherent to iterative schemes, we propose to use a completely different approach, which benefits from the reduced tensor size of the solution. Going back to (13), we know that each instance $\mu \in \mathcal{D}_{\mathcal{I}}$ of the parameter defines a reduced solution $\mathbf{u}_N(\mu) = A_N(\mu)^{-1} \mathbf{f}_N(\mu)$. Since the evaluation of \mathbf{u}_N for small N is cheap, we can find $\mathcal{X}_N \in \mathbb{R}^{n_{\mathcal{I}} \times N}$ from this relation directly via the cross approximation technique from [3]. Note that we can always check the accuracy of the obtained solution a posteriori by computing the residual $\mathcal{R}_N = \mathcal{B}_N - \mathcal{A}_N \mathcal{X}_N$.

6 Numerical Experiments

In this section, numerical experiments illustrate the feasibility and efficiency of the novel *rbTensor* algorithm summarized in Algorithm 4. We study the properties of our methodology for three examples adapted from the literature: the cookie problem from [33], a non-affine elliptic example from [13], and the thermal fin problem from [6, 7, 32].

Algorithm 4 *rbTensor*

- 1: Obtain low-rank tensor approximations of $A(\mu)$ and $f^N(\mu)$ using Algorithm 2 and tensor cross approximation
 - 2: Approximate $u^N(\mu)$ using Algorithm 3 combined with cross approximation for solving linear systems
-

In all examples, we discretize the parameter domain $\mathcal{D} \subset \mathbb{R}^p$ by a tensor grid $\mathcal{D}_{\mathcal{I}} \subset \mathcal{D}$ of 10 Chebyshev points in each parameter direction. For each dimension p , we choose the tree T_D for the hierarchical tensor format $\mathcal{H}_{\mathbf{k}}$ in the form depicted in Figure 3 where the leaf node $\{p+1\}$ corresponds to the physical direction. The subtree $T_{\{1,\dots,p\}}$ is then constructed in a balanced way as in Figure 1 (left).

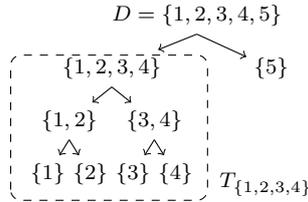


Figure 3: Dimension tree T_D for *rbTensor* for $p = 4$.

For each problem, we construct an approximation of the linear operator A by MDEIM and the approximation of Θ in $\mathcal{H}_{\mathbf{k}}$ as described in Section 4. In particular this means that we ignore any a priori knowledge on the affine structure of the associated bilinear form. In the second step, we greedily construct a reduced basis approximation of the solution u . For each dimension N of the reduced basis space V_N , we solve the projected large tensor system (13) by a black box strategy.

To study the behavior of the reduced basis approximation, we randomly choose a test set $\mathcal{D}_{\text{test}} \subset \mathcal{D}_{\mathcal{I}}$ of 100 sample points for which we report the relative errors of the solution and the residual:

$$\begin{aligned} \text{max_err_sol_rel} &:= \max_{\mu \in \mathcal{D}_{\text{test}}} \frac{\|u^N(\mu) - u_N(\mu)\|_2}{\|u^N(\mu)\|_2} \\ \text{max_err_res_rel} &:= \max_{\mu \in \mathcal{D}_{\text{test}}} \frac{\|A(\mu)u_N(\mu) - f^N(\mu)\|_2}{\|f^N(\mu)\|_2}. \end{aligned}$$

For a given output functional $\ell : X \rightarrow \mathbb{R}$, we measure the relative error:

$$\text{max_err_out_rel} := \max_{\mu \in \mathcal{D}_{\text{test}}} \frac{|\ell(u^N(\mu)) - \ell(u_N(\mu))|}{|\ell(u^N(\mu))|}.$$

To analyze the properties of the low-rank tensor approximation, we determine for each N the maximal ranks and effective ranks in the solution vector u_N and in the output functional ℓ . In each example, we measure the time t_N for the assembly and solution of the full FEM problem of size N , the time t_N for the online evaluation of the reduced basis approximation represented in the low-rank tensor format, and the online time t_ℓ for the evaluation of the output functional ℓ . The timings for the full FEM solution have been obtained by a sparse direct solver from an average over 100 independent runs and for the reduced basis approximation by an average over 10.000 runs.

All numerical experiments have been carried out on a quad-core Intel(R) Xeon(R) CPU E31225 with 3.10GHz. Timings are CPU times for a single core. For the FEM approximation we have used the software library deal.II [4].

6.1 Cookie Problem

Let $\Omega = (0, 1)^2$ and $X = H_0^1(\Omega)$. We consider the cookie problem adapted from [33]. The domain Ω encloses disks $\Omega_1, \dots, \Omega_p$ of radius r which are located at a distance of $2r$ from each other and from the boundary of Ω . An illustration of this setting is shown in Figure 4 for $p = 4$ with $r = 1/10$, for $p = 9$ with $r = 1/14$, and for $p = 16$ with $r = 1/18$.

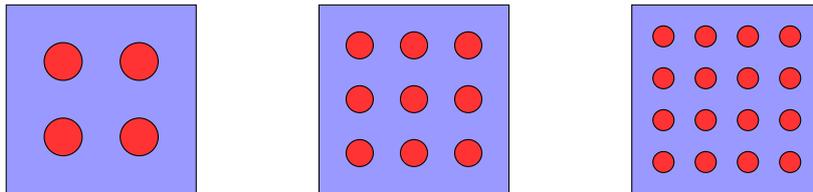


Figure 4: Domain of cookie problem for $p = 4$, $p = 9$, $p = 16$.

Let now $\Omega_0 := \Omega \setminus \bigcup_{i=1}^p \Omega_i$. We consider the variational problem (4) with

$$a(u, v; \mu) = \int_{\Omega_0} \nabla u \cdot \nabla v + \sum_{i=1}^p \mu_i \int_{\Omega_i} \nabla u \cdot \nabla v,$$

right-hand side

$$f(v) = \int_{\Omega} v,$$

and output functional

$$\ell(u) = \int_{\Omega} u. \tag{24}$$

Moreover, we choose $\mathcal{D} := (1, 10)^p$ as a parameter space. Clearly, the bilinear form a is affine in the parameter. In a first sanity test, we check the operator approximation from Section 4 which yields the results in Table 1. We can see that indeed the exact operator rank $Q = p + 1$ is found. Moreover, the effective rank for $\Theta \in \mathbb{R}^{n_T \cdot Q}$ is significantly smaller than the maximal rank. This is due to the fact that for each node $t \in T_D$ one can show that $\text{rank}(\mathcal{M}_t(\Theta)) = \min\{\#t + 1, p + 2 - \#t\}$.

p	ε_{op}	Q	kmax	keff
4	1e-08	5	5	2.69
9	1e-08	10	10	3.81
16	1e-08	17	17	5.14

Table 1: Cooky problem: tensor approximation of $\Theta \in \mathbb{R}^{n_T \cdot Q}$

Once the operator \mathcal{A} is represented in \mathcal{H}_k , we apply our approach in parameter dimensions $p = 4$, $p = 9$, and $p = 16$. The results in Figure 5 indicate that for all p , the relative error in the solution decays relatively fast with increasing reduced basis dimensions N . However, for increasing p , a larger dimension of the reduced basis is needed to obtain a comparable accuracy. As expected, the relative error in the output is always smaller than the error in the solution. Although the maximal ranks in the solution tend to grow proportionally with N , the effective ranks in the solution and in the output remain moderate.

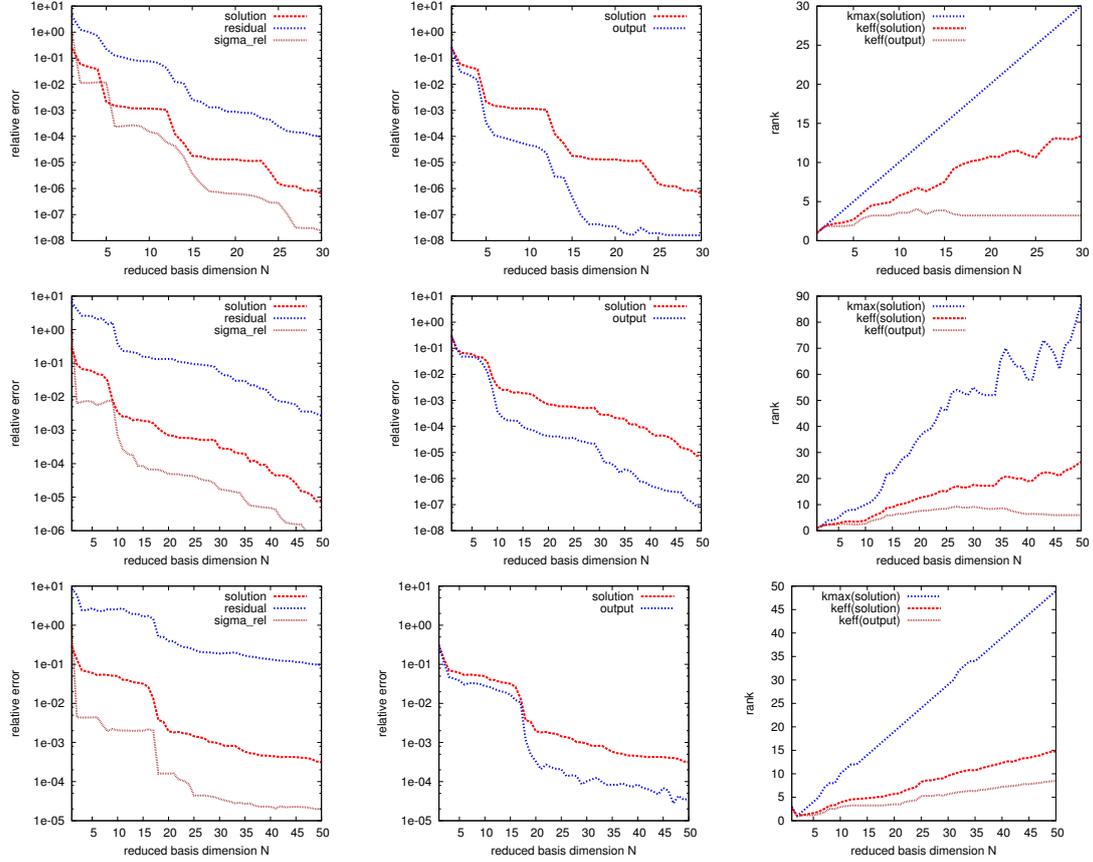


Figure 5: Cooky problem: Top: $p = 4$. Center: $p = 9$. Bottom: $p = 16$. Left: error in the solution and the residual. Center: error in the output functional. Right: tensor ranks for the solution and the output.

Thanks to the low tensor ranks, we observe very low evaluation times both for the full solution u and the output functional ℓ compared to the full FEM simulation as shown in Table 2. In all cases, the evaluation time is dominated by the computation of the matrix-vector product $V_N u_N(\mu)$ which also appears in the standard reduced basis approach.

p	\mathcal{N}	$t_{\mathcal{N}}$ [s]	N	t_N [s]	t_ℓ [s]	t_N/t_N	t_N/t_ℓ	t_{matvec} [s]
4	1169	1.0e-02	10	6.8e-06	1.7e-06	1488	5857	6.3e-06
			20	1.6e-05	7.0e-06	630	1437	9.4e-06
			30	2.7e-05	1.2e-05	374	824	1.5e-05
9	2796	2.6e-02	10	1.5e-05	2.2e-06	1797	11816	1.4e-05
			20	4.6e-05	2.4e-05	568	1093	2.3e-05
			30	9.7e-05	6.3e-05	271	416	3.6e-05
			40	1.2e-04	7.8e-05	216	336	4.5e-05
			50	2.5e-04	2.0e-04	104	130	5.8e-05
16	4769	4.9e-02	10	2.2e-05	4.0e-06	2269	12297	2.3e-05
			20	4.3e-05	6.9e-06	1131	7068	3.8e-05
			30	7.5e-05	2.2e-05	655	2263	5.9e-05
			40	1.1e-04	4.2e-05	430	1158	7.4e-05
			50	1.6e-04	7.1e-05	308	691	9.6e-05

Table 2: Cookie problem: evaluation times $t_{\mathcal{N}}$ for full FEM solution, t_N for online evaluation of u_N , t_ℓ for online evaluation of ℓ compared to t_{matvec} for computing the product $V_N u_N(\mu)$

From the approximation of the output functional ℓ in the hierarchical tensor format, we readily can compute the expected value $\mathbb{E}[\ell]$ and the variance $\mathbb{V}[\ell]$ for a given probability distribution of the parameters $\mu \in \mathcal{D}$. Here we assume that μ is uniformly distributed in \mathcal{D} . To study the relative errors in the expectation and the variance, we use the largest value of N in our computations as a reference value. In Figure 6 we can see that the relative error of the expected value lies below the relative error of the output ℓ whereas the error variance typically lies one or two orders above this error.

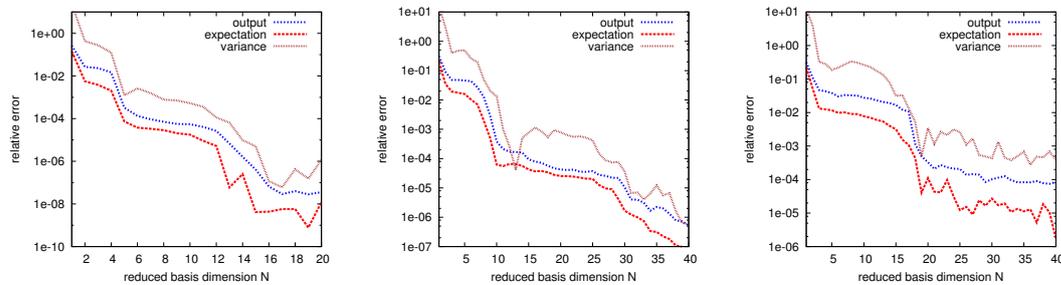


Figure 6: Cookie problem: Relative errors in the expected value $\mathbb{E}[\ell]$ and the variance $\mathbb{V}[\ell]$. Left: $p = 4$. Center: $p = 9$. Right: $p = 16$.

6.2 Non-affine elliptic problem

Let $\Omega = (0, 1)^2$ and $X = H_0^1(\Omega)$. As a second example, we consider a non-affine elliptic problem (4) adapted from [13] with

$$a(u, v; \mu) = a_0(u, v) + a_1(u, v; \mu)$$

where

$$a_0(u, v) := \int_{\Omega} \nabla u \cdot \nabla v, \quad a_1(u, v; \mu) := \int_{\Omega} g(x; \mu) uv,$$

and

$$f(v; \mu) = \int_{\Omega} g(x; \mu) v.$$

Here, $g : \Omega \times \mathcal{D} \rightarrow \mathbb{R}$ is a parameter-dependent function. In [13], one has $g \equiv g_1$, where

$$g_1(x; \mu) := \|(x_1, x_2) - (\mu_1, \mu_2)\|_2^{-1}$$

in parameter dimension $p = 2$. Clearly, g_1 is not a separable function of x and μ such that the bilinear form a_1 does not have an affine representation. More generally, we also consider $g \equiv g_1 + g_2$ and $g \equiv g_1 + g_2 + g_3 + g_4$, with

$$\begin{aligned} g_2(x; \mu) &:= \|(1 - x_1, x_2) - (\mu_3, \mu_4)\|_2^{-1}, \\ g_3(x; \mu) &:= \|(x_1, 1 - x_2) - (\mu_5, \mu_6)\|_2^{-1}, \\ g_4(x; \mu) &:= \|(1 - x_1, 1 - x_2) - (\mu_7, \mu_8)\|_2^{-1} \end{aligned}$$

in parameter dimension $p = 4$, and $p = 8$, respectively. Moreover, let $\mu \in \mathcal{D} := (-0.99, -0.01)^p$ and let the output ℓ be as in (24).

We now ignore our knowledge on the explicit form of the bilinear form a_1 and apply the fully algebraic approximation from Section 4 to the corresponding operator. In Table 3 we can see that for each p the tensor ranks grow with increasing accuracy. Moreover, for larger p also the tensor ranks are higher to reach the same accuracy. Note that the case $p = 2$ corresponds to the results from [13] obtained by the empirical interpolation method.

ε_{op}	$p = 2$			$p = 4$			$p = 8$		
	Q	kmax	keff	Q	kmax	keff	Q	kmax	keff
1e-03	10	10	5.78	15	15	7.44	21	21	8.76
1e-04	14	14	7.36	23	23	11.33	35	35	14.45
1e-05	20	20	9.83	34	34	15.76	49	49	19.84
1e-06	25	25	10.88	44	44	19.81	66	66	26.50

Table 3: Non-affine problem: tensor approximation of $\Theta \in \mathbb{R}^{n_{\mathcal{I}} \cdot Q}$.

To study the effect of the operator approximation on the attainable accuracy in the solution, we apply our algorithm for different operator tolerances to the given problem. In Figure 7 we can see that the accuracy in the solution is indeed only slightly smaller than the prescribed tolerance for the operator.

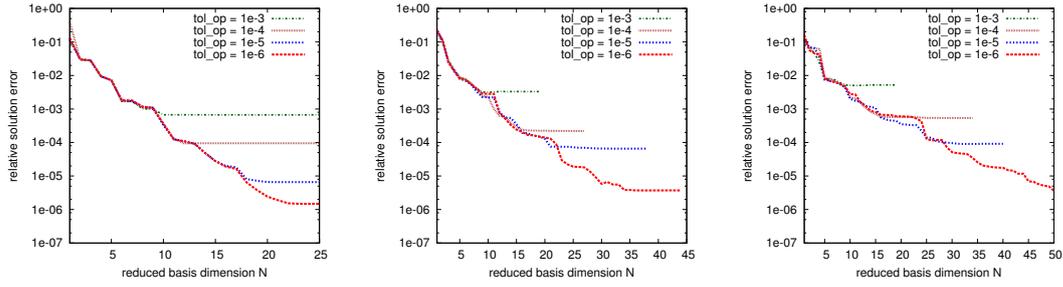


Figure 7: Non-affine problem: Relative errors in the solution depending on the tolerance in the operator approximation. Left: $p = 2$. Center: $p = 4$. Right: $p = 8$.

In the following, we fix the operator tolerance $\varepsilon_{\text{op}} = 10^{-6}$ and apply our approach in parameter dimension $p = 2$, $p = 4$, and $p = 8$. The results in Figure 8 indicate that for all p , the relative error in the solution decays relatively fast with increasing reduced basis dimensions N until the operator tolerance is reached. For $p = 4$ and $p = 8$, the behavior of the errors and of the ranks is quite similar which might be due to the symmetry of the problem.

As before, we observe very low evaluation times both for the full solution u and the output functional ℓ compared to the full FEM simulation as shown in Table 4. Note also that in contrast to [13], the evaluation time for u_N does no longer depend on the accuracy of the operator approximation, i.e., on Q .

p	\mathcal{N}	$t_{\mathcal{N}}$ [s]	N	t_N [s]	t_ℓ [s]	$t_{\mathcal{N}}/t_N$	$t_{\mathcal{N}}/t_\ell$	t_{matvec} [s]
2	4225	3.9e-02	10	2.0e-05	1.5e-06	1925	26314	2.2e-05
			20	3.4e-05	2.8e-06	1131	13566	3.4e-05
4	4225	4.0e-02	10	2.1e-05	2.6e-06	1890	15270	2.2e-05
			20	4.1e-05	9.6e-06	979	4189	3.4e-05
			30	7.1e-05	2.3e-05	566	1740	5.1e-05
			40	1.0e-04	3.9e-05	399	1027	6.4e-05
8	4225	4.3e-02	10	2.4e-05	5.4e-06	1811	7912	2.2e-05
			20	5.2e-05	2.0e-05	823	2115	3.4e-05
			30	1.1e-04	5.6e-05	401	769	5.1e-05
			40	1.8e-04	1.1e-04	241	375	6.4e-05
			50	3.0e-04	2.1e-04	141	201	8.3e-05

Table 4: Non-affine problem: evaluation times $t_{\mathcal{N}}$ for full FEM solution, t_N for online evaluation of u_N , t_ℓ for online evaluation of ℓ with $\varepsilon_{\text{op}} = 10^{-6}$ compared to t_{matvec} for computing the product $V_N u_N(\mu)$

6.3 Thermal Fin

As a third example, we consider the thermal fin problem adapted from [6, 7, 32]. In this model, one studies a cooling device $\Omega \subset \mathbb{R}^2$ of the form shown in Figure 9 with a heat source at the bottom boundary Γ^{root} that dissipates heat through the domain Ω to the ambient space. The domain Ω consist of n_{fin} subfins as depicted in Figure 9 (right) for $n_{\text{fin}} = 1, 2, 3$.

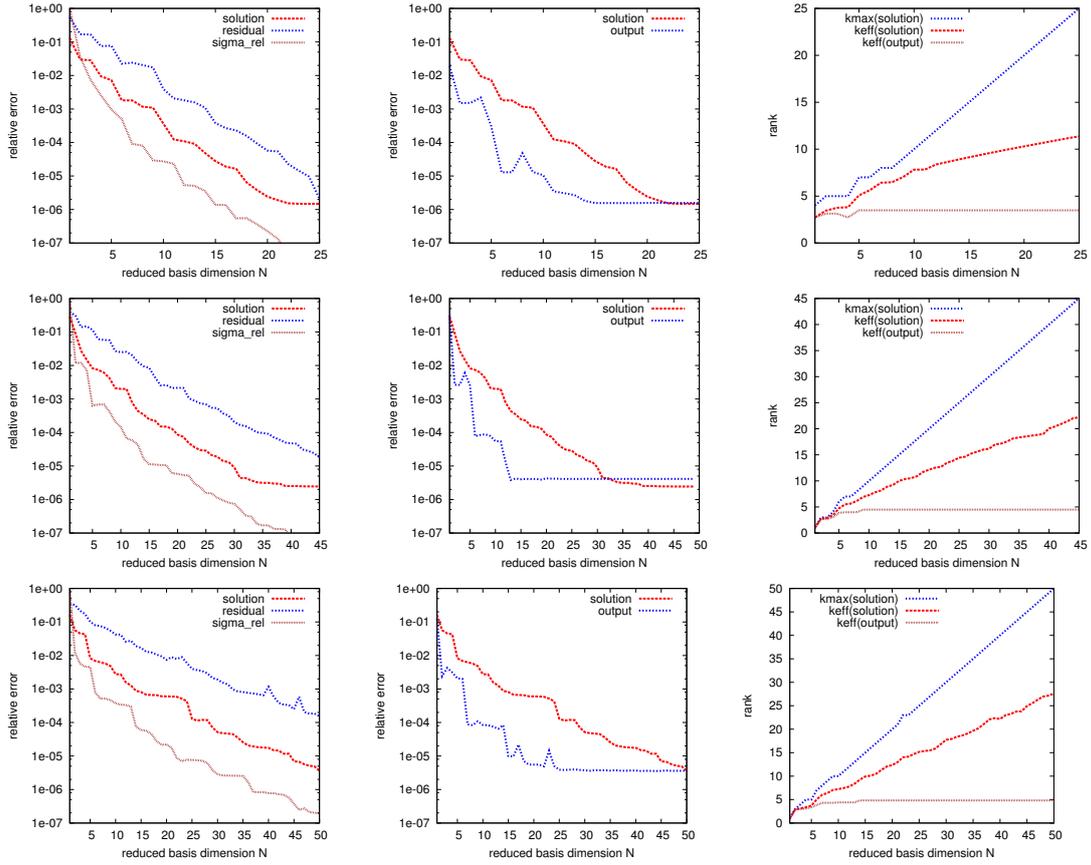


Figure 8: Non-affine problem: Top: $p = 4$. Center: $p = 9$. Bottom: $p = 16$. Left: error in the solution and the residual. Center: error in the output functional. Right: tensor ranks for the solution and the output.

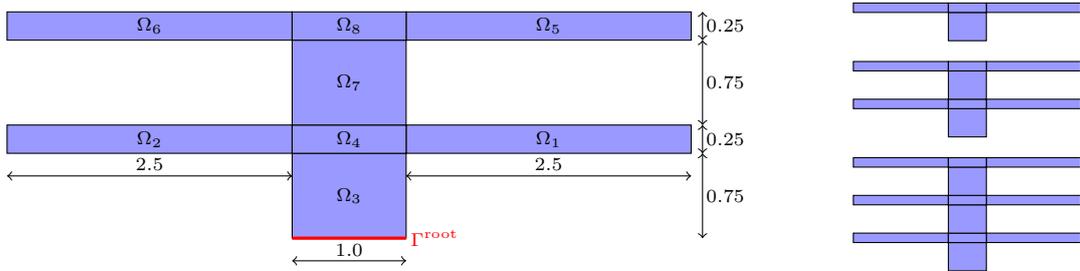


Figure 9: Left: Reference domain Ω_{ref} for fin with 2 subfins. Right: Fin with $n_{\text{fin}} = 1, 2, 3$ subfins

Each subfin $s = 1, \dots, n_{\text{fin}}$ consists of four subdomains $\Omega_{j+1}, \dots, \Omega_{j+4}$ with $j := 4(s-1)$. On each subdomain Ω_i , $i = 1, \dots, 4n_{\text{fin}}$, the thermal conductivity κ_i is assumed

to be constant. The steady-state temperature distribution u is given by

$$-\kappa_i \Delta u = 0 \quad \text{in } \Omega_i, \quad i = 1, \dots, 4n_{\text{fin}}.$$

At the interior boundaries $\Gamma_{i,j}^{\text{int}} := \partial\Omega_i \cap \partial\Omega_j$, $i \neq j$, we impose the interface conditions

$$\begin{aligned} u|_{\Omega_i} &= u|_{\Omega_j} \quad \text{on } \Gamma_{i,j}^{\text{int}}, \\ -\kappa_i(\partial u / \partial n_i) &= -\kappa_j(\partial u / \partial n_j) \quad \text{on } \Gamma_{i,j}^{\text{int}}, \end{aligned}$$

where n_i, n_j are the outer normals of Ω_i, Ω_j , respectively. On the exterior boundaries $\Gamma_i^{\text{ext}} := \partial\Omega_i \setminus \bigcup_j \Gamma_{i,j}^{\text{int}}$, we model the convective heat loss by Robin boundary conditions of the form

$$-\kappa_i(\partial u / \partial n_i) = \text{Bi } u \quad \text{on } \Gamma_i^{\text{ext}}$$

where Bi is the Biot number. Finally, we impose a heat source at the bottom boundary Γ^{root} by the Neumann condition

$$-\kappa_3(\partial u / \partial n_3) = -1 \quad \text{on } \Gamma^{\text{root}}.$$

The variational formulation of this problem is then given by (4) with $X = H^1(\Omega)$ and

$$a(u, v) := \sum_{i=1}^{4n_{\text{fin}}} \kappa_i \int_{\Omega_i} \nabla u \cdot \nabla v + \text{Bi} \sum_{i=1}^{4n_{\text{fin}}} \int_{\Gamma_i^{\text{ext}}} uv$$

and

$$f(v) := \int_{\Gamma^{\text{root}}} v.$$

We now parametrize this model by a parameter vector $\mu \in \mathcal{D} \subset \mathbb{R}^p$ with $p := 4n_{\text{fin}} + 3$ in the following way:

1. For each subfin $s = 1, \dots, n_{\text{fin}}$, we parametrize the heat conductivity of the domains Ω_{j+1} and Ω_{j+2} , $j := 4(s-1)$, by setting $\kappa_{j+1} = \kappa_{j+2} := \mu_{j+1}$. For the rest of the subdomains, we assume an identical conductivity $\kappa_{j+3} = \kappa_{j+4} := \mu_{4n_{\text{fin}}+1}$.
The range of the parameters is given by $\mu_{j+1}, \mu_{4n_{\text{fin}}+1} \in [0.5, 2.0]$.
2. For each subfin $s = 1, \dots, n_{\text{fin}}$, we parametrize the shape of the domains Ω_{j+1} and Ω_{j+2} , $j := 4(s-1)$, to be rectangles of size $2.5\mu_{j+2} \times 0.25\mu_{j+3}$. Moreover, Ω_{j+3} is assumed to be a rectangle of size $\mu_{4n_{\text{fin}}+2} \times 0.75\mu_{j+4}$ which makes Ω_{j+4} a rectangle of size $\mu_{4n_{\text{fin}}+2} \times 0.25\mu_{j+3}$.
The range of the parameters is given by $\mu_{j+2}, \mu_{j+3}, \mu_{4n_{\text{fin}}+2} \in [0.8, 1.2]$.
3. We set the Biot number to $\text{Bi} := \mu_{4n_{\text{fin}}+3}$.
The range of the parameter is given by $\mu_{4n_{\text{fin}}+3} \in [0.5, 2]$.

Thanks to the Cartesian product structure of the domain, the parametrized problem can be easily transformed to a variational problem on a reference domain Ω_{ref} depicted

p	ε_{op}	Q	kmax	keff
7	1e-08	10	10	4.95
11	1e-08	19	19	6.93
15	1e-08	28	28	8.52

Table 5: Thermal fin: tensor approximation of $\Theta \in \mathbb{R}^{n_T \times Q}$

in Figure 9 (left) for the case $n_{\text{fin}} = 2$. For the details we refer to [6]. As an output functional, we again choose ℓ from (24).

It can be shown that the linear operator associated to the bilinear form a on the reference domain again allows for an affine representation. This structure is automatically found by our algorithm, cf. Table 5.

The results in Figure 10 illustrate that the dimension N of the reduced basis needs to be chosen significantly larger to reach the same accuracy when the complexity of the problem is increased (larger n_{fin}). For a fixed dimension N , the tensor ranks of the solution grow only mildly with increasing n_{fin} . Note however that for a fixed accuracy, also the required tensor ranks are higher. The computational savings in time shown in Table 6 show a similar behavior as in the previous examples.

p	\mathcal{N}	$t_{\mathcal{N}}$ [s]	N	t_N [s]	t_ℓ [s]	$t_{\mathcal{N}}/t_N$	$t_{\mathcal{N}}/t_\ell$	t_{matvec} [s]
7	2571	1.7e-02	10	1.8e-05	6.7e-06	956	2604	1.3e-05
			20	4.3e-05	2.2e-05	404	796	2.2e-05
			30	8.2e-05	5.0e-05	214	349	3.3e-05
			40	1.2e-04	7.9e-05	145	221	4.2e-05
			50	2.5e-04	2.0e-04	69	86	5.3e-05
11	5747	4.4e-02	10	3.8e-05	1.3e-05	1152	3405	2.8e-05
			20	1.2e-04	8.1e-05	355	540	4.6e-05
			30	1.8e-04	1.1e-04	249	416	7.1e-05
			40	3.2e-04	2.1e-04	139	205	9.1e-05
			50	5.6e-04	4.0e-04	78	110	1.2e-04
15	9375	8.0e-02	10	5.7e-05	1.5e-05	1402	5328	4.7e-05
			20	1.7e-04	9.2e-05	482	873	7.7e-05
			30	4.8e-04	3.0e-04	168	266	1.2e-04
			40	5.4e-04	3.0e-04	148	265	1.6e-04
			50	1.1e-03	6.7e-04	73	119	2.1e-04

Table 6: Thermal fin: evaluation times $t_{\mathcal{N}}$ for full FEM solution, t_N for online evaluation of u_N , and t_ℓ for online evaluation of ℓ compared to t_{matvec} for computing the product $V_N u_N(\mu)$

7 Conclusions

We have proposed a new method, called *rbTensor*, for the efficient solution of parametrized linear systems. Combining techniques from reduced basis methods and low-rank tensor approximation, our method is particularly suited in the presence of many parameters and nonlinear dependencies. A major advantage of *rbTensor* compared to existing low-rank tensor methods is that little information on the problem structure is needed, thanks to a combination of black box approximation techniques combined with dimension reduction.

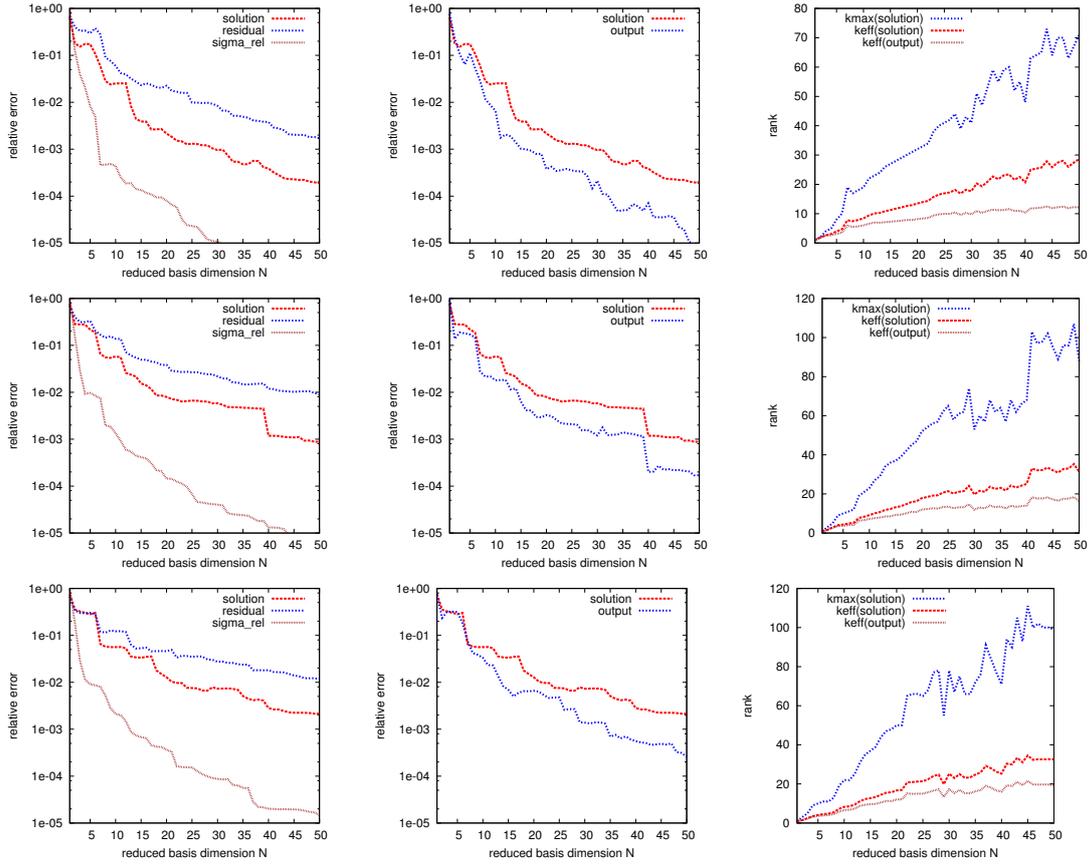


Figure 10: Non-affine problem: Top: $p = 7$. Center: $p = 11$. Bottom: $p = 15$. Left: error in the solution and the residual. Center: error in the output functional. Right: tensor ranks for the solution and the output.

Compared to reduced basis methods, *rbTensor* can cope with very fine discretizations of the (continuous) parameter domain \mathcal{D} , potentially allowing for a finer control of the error on \mathcal{D} . More importantly, parametrized solutions can be directly accessed via queries to the solution tensor, which not only allows for fast online evaluation but also for other operations, such as optimization and the computation of statistics of the solution.

The advantages of *rbTensor* come at the expense of a more elaborate offline phase. Moreover, black box approximation is not entirely reliable, which may lead to erroneous results, even though we have never observed this in our numerical experiments. The combination of low-rank tensor techniques with error estimates based on inf-sup constants [20] remains an open problem.

References

- [1] J. Ballani and L. Grasedyck. A projection method to solve linear systems in tensor format. *Numer. Linear Algebra Appl.*, 20(1):27–43, 2013.
- [2] J. Ballani and L. Grasedyck. Hierarchical tensor approximation of output quantities of parameter-dependent PDEs. Preprint, EPFL-SB-MATHICSE-ANCHP, 2014.
- [3] J. Ballani, L. Grasedyck, and M. Kluge. Black box approximation of tensors in hierarchical Tucker format. *Linear Algebra Appl.*, 438(2):639–657, 2013.
- [4] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.
- [5] M. Bebendorf, Y. Maday, and B. Stamm. Comparison of some reduced representation approximations. In A. Quarteroni and G. Rozza, editors, *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *Modeling, Simulation and Applications*, pages 67–100. Springer International Publishing, 2014.
- [6] T. Bui-Thanh. *Model-constrained optimization methods for reduction of parameterized large-scale systems*. PhD thesis, MIT, 2007.
- [7] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM J. Sci. Comput.*, 30(6):3270–3288, 2008.
- [8] S. Dolgov. TT-GMRES: solution to a linear system in the structured tensor format. *Russian J. Numer. Anal. Math. Modelling*, 28(2):149–172, 2013.
- [9] S. Dolgov and I. V. Oseledets. Solution of linear systems and matrix inversion in the TT-format. *SIAM J. Sci. Comput.*, 34(5):A2718–A2739, 2012.
- [10] S. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM J. Sci. Comput.*, 36(5):A2248–A2271, 2014.
- [11] L. Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.*, 31:2029–2054, 2010.
- [12] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.
- [13] 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.
- [14] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer, Berlin, 2012.

- [15] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *J. Fourier Anal. Appl.*, 15(5):706–722, 2009.
- [16] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, Berlin, 2016.
- [17] J. S. Hesthaven, B. Stamm, and S. Zhang. Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(1):259–283, 2014.
- [18] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, second edition, 2002.
- [19] S. Holtz, T. Rohwedder, and R. Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM J. Sci. Comput.*, 34(2):A683–A713, 2012.
- [20] D. B. P. Huynh, D. J. Knezevic, Y. Chen, J. S. Hesthaven, and A. T. Patera. A natural-norm successive constraint method for inf-sup lower bounds. *Comput. Methods Appl. Mech. Engrg.*, 199(29-32):1963–1975, 2010.
- [21] 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.
- [22] 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.
- [23] D. Kressner, M. Plešinger, and C. Tobler. A preconditioned low-rank CG method for parameter-dependent Lyapunov matrix equations. *Numer. Linear Algebra Appl.*, 21(5):666–684, 2014.
- [24] D. Kressner, M. Steinlechner, and B. Vandereycken. Preconditioned low-rank Riemannian optimization for linear systems with tensor product structure. Preprint, EPFL-SB-MATHICSE-ANCHP, 2015.
- [25] 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.
- [26] H. G. Matthies and E. Zander. Solving stochastic systems with low-rank tensor compression. *Linear Algebra Appl.*, 436(10):3819–3838, 2012.
- [27] 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.

- [28] F. Negri, A. Manzoni, and D. Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. Preprint 2/2015, EPFL-SB-MATHICSE, 2015.
- [29] I. V. Oseledets. DMRG approach to fast linear algebra in the TT-format. *Comput. Meth. Appl. Math*, 11(3):382–393, 2011.
- [30] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations*. Springer, Unitext Series, vol. 92, 2016.
- [31] 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.
- [32] Y. O. Solodukhov. *Reduced-basis methods applied to locally non-affine and locally non-linear partial differential equations*. PhD thesis, MIT, 2005.
- [33] C. Tobler. *Low-rank Tensor Methods for Linear Systems and Eigenvalue Problems*. PhD thesis, ETH Zürich, 2012.

Recent publications:

MATHEMATICS INSTITUTE OF COMPUTATIONAL SCIENCE AND ENGINEERING
Section of Mathematics
Ecole Polytechnique Fédérale
CH-1015 Lausanne

- 15.2015** ANDREA MANZONI, STEFANO PAGANI:
A certified reduced basis method for PDE-constrained parametric optimization problems by an adjoint-based approach
- 16.2015** SIMONE DEPARIS, DAVIDE FORTI, ALFIO QUARTERONI:
A fluid-structure interaction algorithm using radial basis function interpolation between non-conforming interfaces
- 17.2015** ASSYR ABDULLE, ONDREJ BUDAC:
A reduced basis finite element heterogeneous multiscale method for Stokes flow in porous media
- 18.2015** DANIEL KRESSNER, MICHAEL STEINLECHNER, BART VANDEREYCKEN:
Preconditioned low-rank Riemannian optimization for linear systems with tensor product structure
- 19.2015** ALESSANDRO S. PATELLI, LUCA DEDÈ, TONI LASSILA, ANDREA BARTEZZAGHI, ALFIO QUARTERONI:
Isogeometric approximation of cardiac electrophysiology models on surfaces: an accuracy study with application to the human left atrium
- 20.2015** MATTHIEU WILHELM, LUCA DEDÈ, LAURA M. SANGALLI, PIERRE WILHELM:
IGS: an IsoGeometric approach for Smoothing on surfaces
- 21.2015** SIMONE DEPARIS, DAVIDE FORTI, PAOLA GERVASIO, ALFIO QUARTERONI:
INTERNODES: an accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces
- 22.2015** ABDUL-LATEEF HAJI-ALI, FABIO NOBILE, LORENZO TAMELLINI, RAÛL TEMPONE:
Multi-index stochastic collocation for random PDEs
- 23.2015** SIMONE BRUGIAPAGLIA, FABIO NOBILE, STEFANO MICHELETTI, SIMONA PEROTTO:
A theoretical study of COmpressed SolvING for advection-diffusion-reaction problems
- 24.2015** ANA ŠUŠNJARA, NATHANAËL PERRAUDIN, DANIEL KRESSNER, PIERRE VANDERGHEYNST:
Accelerate filtering on graphs using Lanczos method
- 25.2015** FRANCESCO BALLARIN, ELENA FAGGIANO, SONIA IPPOLITO, ANDREA MANZONI, ALFIO QUARTERONI, GIANLUIGI ROZZA, ROBERTO SCROFANI:
Fast simulation of patient-specific haemodynamics of coronary artery bypass grafts based on a Pod-Galerkin method and a vascular shape parametrization
- 26.2015** FRANCISCO MACEDO:
Benchmark problems on stochastic automata networks in tensor train format
- 27.2015** JONAS BALLANI, DANIEL KRESSNER :
Reduced basis methods: from low-rank matrices to low-rank tensors