SPECTRAL TENSOR-TRAIN DECOMPOSITION

DANIELE BIGONI†, ALLAN P. ENGSIG-KARUP†, AND YOUSSEF M. MARZOUK‡

Abstract. The accurate approximation of high-dimensional functions is an essential task in uncertainty quantification and statistical inference. We propose a new function approximation scheme based on a spectral extension of the tensor-train (TT) decomposition. We first define a functional version of the TT decomposition and analyze its properties. We obtain results on the convergence of the decomposition, revealing links between the regularity of the function, the dimension of the input space, and the TT ranks. We also show that the regularity of the target function is preserved by the univariate functions (i.e., the “cores”) comprising the functional TT decomposition. This result motivates an approximation scheme employing polynomial approximations of the cores. For functions with appropriate regularity, the resulting spectral tensor-train decomposition combines the linear dimension-scaling of the TT decomposition with the spectral convergence rate of polynomial approximations, yielding efficient and accurate surrogates for high-dimensional functions. To construct these decompositions, we use the sampling algorithm TT-DMRG-cross to obtain the TT decomposition of tensors resulting from suitable discretizations of the target function, and we show that the required number of function evaluations scales linearly with dimension. We assess the performance of the method on a range of numerical examples: Genz functions of dimension up to 100, a modified set of Genz functions for which the difficulty of the approximation does not decrease with dimension, and functions with mixed Fourier modes or with local features. The method is also used to approximate the solution of an elliptic PDE with random input data, where the diffusivity is modeled by a log-normal random field. The software and examples presented in this work are available online.

Key words. Approximation theory, tensor-train decomposition, orthogonal polynomials, uncertainty quantification.

AMS subject classifications. 41A10, 41A63, 41A65, 46M05, 65D15

1. Introduction. High-dimensional functions appear frequently in science and engineering applications, where a quantity of interest may depend in nontrivial ways on a large number of independent variables. In the field of uncertainty quantification, for example, stochastic partial differential equations (PDEs) are often characterized by hundreds or thousands of independent stochastic parameters. A numerical approximation of the PDE solution must capture the coupled effects of all these parameters on the entire solution field, or on any quantity of interest that is a functional of the solution field. Problems of this kind quickly become intractable when confronted with naïve approximation methods, and the development of more effective methods is a long-standing challenge. This paper develops a new approach for high-dimensional function approximation, combining the discrete tensor-train format with spectral theory for polynomial approximation.

For simplicity, we will focus on real-valued functions representing the parameter dependence of a single quantity of interest. For a function $f \in L^2([a,b]^d)$, a straightforward approximation might involve projecting $f$ onto the space spanned by the tensor product of basis functions $\{\phi_{i_1}(x_1) \otimes \cdots \otimes \phi_{i_d}(x_d)\}_{i_1,\cdots,i_d=1}^{n_1,\cdots,n_d} \subset L^2([a,b])$ for $j=1\ldots d$, obtaining:

$$f \simeq \sum_{i_1}^{n_1} \cdots \sum_{i_d}^{n_d} c_{i_1,\ldots,i_d} (\phi_{i_1} \otimes \cdots \otimes \phi_{i_d}).$$

This approach quickly becomes impractical as the parameter dimension $d$ increases, due to the exponential growth in the number of coefficients $c_{i_1,\ldots,i_d}$ and the computational effort (i.e., the number of function evaluations) required to determine their values. This growth is a symptom of the “curse of dimensionality.”

Attempts to mitigate the curse of dimensionality typically employ some assumption about the structure of the function under consideration, effectively reducing the number of coefficients that
must be computed. A widely successful class of methods involves interpolation or pseudospectral approximation with sparse grids [1, 61, 42, 11, 10]: instead of taking a full tensor product approximation as in (1.1), one considers a Smolyak sum [54] of smaller full-tensor approximations, each perhaps involving only a subset of the input parameters or at most low-order interactions among all the inputs. While the basis functions \( \phi_i \) are typically selected \emph{a priori}, the components of the Smolyak sum can be chosen adaptively [10]. In general, these approaches work best when inputs to the target function \( f \) are weakly coupled.

Other approaches to high-dimensional approximation rely on \emph{low-rank separated representations}, e.g., of the form:

\[
(1.2) \quad f \approx \sum_{i=1}^{r} c_i \gamma_{i,1} \otimes \cdots \otimes \gamma_{i,d},
\]

where \( \gamma_{i,1}, \ldots, \gamma_{i,d} : [a,b] \to \mathbb{R} \), for \( i = 1 \ldots r \), and \( r \) is ideally small (hence, the descriptor ‘low-rank’). In some cases, the chosen representation might separate only certain blocks of inputs to \( f \), e.g., spatial and stochastic variables [43, 55, 10]. In general, however, inputs to \( f \) can all be separated as in (1.2). The representation in (1.2) is analogous to the \emph{canonical decomposition} of a tensor [34]. Many strategies for constructing low-rank separated representations of parameterized models have been developed [11, 38, 32, 33, 44, 43, 9, 11, 14, 15, 12, 62]; these include the proper generalized decomposition [43, 55, 9], least-squares approaches [13], and tensor-structured Galerkin approximations [33, 41, 15]. Almost all of these approaches are “intrusive” in the sense that they require access to more than black-box evaluations of the target function \( f \). But non-intrusive approaches have recently been developed as well [14].

An alternative to the canonical tensor decomposition is the \emph{tensor-train} (TT) format for discrete tensors, introduced by [45]. As we will describe in Section 2 the TT format offers a number of advantages over the canonical decomposition, and it is therefore attractive to consider its application to function approximation. Recent work employing TT in the context of uncertainty quantification includes [38], which uses the TT format to compress the operator and the polynomial coefficients arising in the stochastic Galerkin discretization of an elliptic PDE. In [32] the \emph{quantics tensor-train} (QTT) format is used to accelerate the preconditioned iterative solution of multiparametric elliptic PDEs. [62] uses TT-cross interpolation [46] to evaluate the three-term recurrence relation used to find orthogonal polynomials and Gaussian quadrature points for arbitrary probability measures. [15] compares the TT format with the canonical decomposition and the hierarchical Tucker decomposition, for the purpose of storing the operator derived from the Galerkin discretization of a stochastic PDE, and for computing the associated inner products. While these efforts use the TT format to achieve important efficiency gains in solving particular UQ problems, they do not address the general non-intrusive function approximation problem considered in this paper.

In this work, we will use classical polynomial approximation theory to extend the \emph{discrete} TT decomposition into a scheme for the approximation of continuous functions. To do this, we will first construct the \emph{functional} counterpart of the tensor-train decomposition and examine its convergence. We will prove that the functional TT decomposition converges for a wide class of functions in \( L^2 \) that satisfy a particular regularity condition; this result highlights connections between the regularity of the target function, the dimension of the input space, and the TT ranks. For this class of functions, we will also show that the weak differentiability of the target function is preserved by the univariate functions or “cores” comprising the functional TT decomposition, allowing us to apply polynomial approximation theory to the latter. The resulting combined \emph{spectral} TT approximation exploits the regularity of the target function \( f \) and converges exponentially for smooth functions, but yields a representation whose complexity scales linearly with dimension.

We will focus on the non-intrusive setting where \( f \) is a black-box function that can only be evaluated at chosen parameter values. Hence we must resort to a sampling method in constructing the spectral TT approximation: we will use the rank-revealing TT-DMRG-cross technique [48] to
approximate the tensors resulting from suitable discretizations of $f$. We will then assess the performance of the spectral TT approximation on a range of target functions, including the Genz test functions, functions with Fourier spectra chosen to illustrate particular challenges, functions with local features, and functions induced by the solution of a stochastic elliptic PDE. In all these examples, we will comment on the relationships between the degree of the polynomial approximation, the TT ranks, the accuracy of the overall approximation, and the scaling of computational effort with dimension.

The remainder of the paper is organized as follows. In Section 2 we recall the definitions and properties of several tensor decomposition formats, focusing on the TT decomposition. Section 3 reviews relevant results on the approximation of functions in Sobolev spaces. In Section 4, we provide a constructive definition of the functional TT decomposition, discuss its convergence, and present results on the regularity of the decomposition. This leads to algorithms for constructing the spectral TT decomposition, whose practical implementations are summarized in Section 4.3.3. Section 5 presents the numerical examples. Some technical results are deferred to the Appendix.

2. Tensor decompositions. For the moment, assume that we can afford to evaluate the function $f : [a, b]^d \to \mathbb{R}$ at all points on a tensor grid $X = \prod_{j=1}^d x_j$, where $x_j = (x_{ij})_{i_j=1}^{n_j}$ for $j = 1, \ldots, d$. We denote $A(i_1, \ldots, i_d) = f(x_{i_1}, \ldots, x_{i_d})$ and abbreviate the $d$-dimensional tensor by $A = f(X)$.

In the special case of $d = 2$, $A$ reduces to a matrix $A$. The singular value decomposition (SVD) of this matrix,

\begin{equation}
A = U \Sigma V^T,
\end{equation}

always exists and, since $A$ is a real-valued matrix, is unique up to sign changes \[57\]. The SVD can be used to obtain a low-rank approximation of $A$ by truncating away the smallest singular values on the diagonal of $\Sigma$ and the corresponding columns of $U$ and $V$. Unfortunately the SVD cannot be immediately generalized to tensors of dimension $d > 2$. Several approaches to this problem have been proposed over the years \[34, 5, 24\]. By far the most popular are the canonical decomposition (CANDECOMP) \[7, 28\], the Tucker decomposition \[58\], and the tensor-train decomposition \[45\].

2.1. Classical tensor decompositions. The canonical decomposition aims to represent $A$ as a sum of outer products:

\begin{equation}
A \simeq A_{CD} = \sum_{i=1}^r A_{i(1)} \circ \cdots \circ A_{i(d)},
\end{equation}

where $A_{i(k)}$ is the $i$-th column of matrix $A^{(k)} \in \mathbb{R}^{n_k \times r}$ and $\circ$ denotes the outer product of two vectors. The upper bound of summation $r$ is called the canonical rank of the tensor $A_{CD}$. The canonical decomposition is unique under mild conditions \[51\]. On the other hand a best rank-$r$ decomposition—where one truncates the expansion similarly to the SVD—does not always exist since the space of rank-$r$ tensors is not closed \[37, 52\]. Computation of the canonical decomposition based on the alternating least squares (ALS) method is not guaranteed to find a global minimum of the approximation error, and has a number of other drawbacks and corresponding workarounds \[34\].

The Tucker decomposition is defined as follows:

\begin{equation}
A \simeq \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} g_{i_1 \ldots i_d} \left( A_{i_1(1)} \circ \cdots \circ A_{i_d(d)} \right),
\end{equation}

where the core tensor $G$, defined by $G(i_1, \ldots, i_d) = g_{i_1 \ldots i_d}$, weighs interactions between different components in different dimensions. This expansion is not unique, due to the possibility of applying
a rotation to the core tensor and its inverse to the components $A^{(i)}$. However, the ability to recover a unique decomposition can be improved if some sparsity is imposed on the core tensor $\mathbf{G}$. The Tucker decomposition does not suffer from the same closure problem as the canonical decomposition, but the number of parameters to be determined grows exponentially with the dimension $d$ due to the presence of the core tensor $\mathbf{G}$. This cost limits the applicability of Tucker decomposition to relatively low-dimensional problems.

### 2.2. Tensor-train decomposition

The dimension limitations of the Tucker decomposition can be overcome using a hierarchical singular value decomposition, where the tensor is not decomposed with a single core $\mathbf{G}$ that simultaneously relates all the dimensions, but rather with a hierarchical tree of cores—usually binary—that relate a few dimensions at a time. This approach is called the hierarchical Tucker or $\mathcal{H}$-Tucker decomposition [23]. A particular type of $\mathcal{H}$-Tucker decomposition is the tensor-train decomposition, which retains many of the characteristics of the $\mathcal{H}$-Tucker decomposition but with a simplified formulation. (See [23, Sec. 5.3] for a comparison.) The tensor-train decomposition has the following attractive properties:

- existence of the full-rank approximation [45, Thm. 2.1],
- existence of the low-rank best approximation [45, Cor. 2.4],
- an algorithm that returns a sub-optimal TT-approximation (see (2.7) and [45, Cor. 2.4]),
- memory complexity that scales linearly with dimension $d$ [45, Sec. 3],
- straightforward multi-linear algebra operations, and
- a sampling algorithm for constructing the TT-approximation, with a computational complexity that scales linearly with the dimension $d$ [48].

**Definition 2.1 (Discrete tensor-train approximation).** Let $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ have entries $A(i_1, \ldots, i_d)$. The $\text{TT-rank-}r = (r_0, \ldots, r_d)$ approximation of $A$ is $A_{TT} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, defined as:

$$
A(i_1, \ldots, i_d) = A_{TT}(i_1, \ldots, i_d) + E_{TT}(i_1, \ldots, i_d)
$$

(2.4)

where $E_{TT}$ is the residual term and $r_0 = r_d = 1$.

The three-dimensional arrays $G_k(\alpha_k, i_k, \alpha_{k+1})$ are referred to as TT cores. The TT format approximates every entry of the tensor $A$ with a product of matrices, in particular with a sequence of $r_k \times r_{k+1}$ matrices, each indexed by the parameter $i_{k+1}$. In other words, each core $G_k$ is “connected” to the adjacent cores $G_{k-1}$ and $G_{k+1}$ by summing over the indices $\alpha_{k-1}$ and $\alpha_k$; hence the name tensor ‘train.’ It can be shown [45] that the TT-approximation is exact ($E_{TT} = 0$) when

$$
r_k = \text{rank}(A_k), \quad \forall k \in \{1, \ldots, d\},
$$

(2.5)

where $A_k$ is the $k$-th unfolding of $A$, corresponding to the MATLAB/NumPy operation:

$$
A_k = \text{reshape} \left( A, \prod_{s=1}^{k} n_s, \prod_{s=k+1}^{d} n_s \right).
$$

(2.6)

Furthermore if $r_k \leq \text{rank}(A_k)$, the TT-rank- $r$ approximation $A^{\text{best}}$, which is optimal in the Frobenius norm, always exists and the algorithm TT-SVD [45] produces a quasi-optimal approximation to it. In particular, if $A_{TT}$ is the numerical approximation of $A$ obtained with TT-SVD, then

$$
\|A - A_{TT}\|_F \leq \sqrt{d-1}\|A - A^{\text{best}}\|_F.
$$

(2.7)
With the appropriate scaling of the target tolerance $\delta = \frac{\varepsilon}{\sqrt{d-1}}$, the TT-SVD is then able to construct the approximation $\mathbf{A}_{TT}$ such that
\begin{equation}
\|\mathbf{A} - \mathbf{A}_{TT}\|_F \leq \varepsilon \|\mathbf{A}\|_F .
\end{equation}
Assuming that the TT-ranks are all equal, $r_k = \text{rank}(\mathbf{A}_k) = r$, and that $n_1 = \ldots = n_d = n$, the TT-decomposition $\mathbf{A}_{TT}$ requires the storage of $O\left(dn^2\right)$ parameters. Thus the memory complexity of the representation \eqref{eq:spectraltt} scales linearly with dimension. A further reduction in the required storage can be achieved using the quantics-TT format \cite{47,31} which, for $n = 2^m$, leads to $O\left(dm^2\right)$ complexity.

The computational complexity of the TT-SVD depends on the selected accuracy, but for $r_k = r$ and $n_1 = \ldots = n_d = n$, the algorithm requires $O\left(rn^d\right)$ flops. We see that this complexity grows exponentially with dimension and thus the curse of dimensionality is not resolved, except for the memory complexity of the final compressed representation. At this stage, it is worth noting that using the tensor-train format rather than the more complex $\mathcal{H}$-Tucker decomposition relinquishes the possibility of implementing a parallel version of TT-SVD \cite{49} and gaining a factor of $1/\log_2(d)$ in computational complexity. But this would still not resolve the exponential growth of computational complexity with respect to dimension. Another reason that the TT-SVD may not be immediately suitable for high-dimensional problems is that it first requires storage of the full tensor. This means that the initial memory requirements scale exponentially with the problem’s dimension. In the next section we will discuss an alternative method for constructing a TT approximation of the tensor using a small number of function evaluations.

An open question in tensor-train decomposition regards the ordering of the $d$ indices of $\mathbf{A}$; different orderings can lead to higher or lower TT-ranks, and change the memory efficiency of the representation accordingly. Given a particular permutation $\sigma$, we define the re-ordered tensor $\mathbf{B}(i) = \mathbf{A}(\sigma(i))$. One would like to find $\sigma$ such that the TT-ranks of $\mathbf{B}$ are minimized. From \eqref{eq:tt-ranks} we see that the TT-ranks depend on the ranks of the unfoldings $\mathbf{B}_k$ of $\mathbf{B}$, and from the definition of the unfolding \eqref{eq:tt-unfolding} one sees that two indices $i$ and $j$ will influence the ranks of the matrices $\{\mathbf{B}_k\}_{k=i}^{j-1}$. The permutation $\sigma$ should be chosen such that pairs of indices yielding high-rank unfoldings are contiguous, so that the rank will be high only for a limited number of unfoldings. If this does not happen, the non-separability of pairs of dimensions is carried from core to core, making the decomposition more expensive. Section \ref{sec:5.3} will point to several examples where this problem arises.

### 2.3. Cross-interpolation of tensors

An alternative to TT-SVD is provided by the TT-DMRG-cross algorithm. (See \cite{48} for a detailed description.) This method hinges on the notion of the density matrix renormalization group \cite{60} (DMRG) and on matrix skeleton decomposition \cite{22}. For $d = 2$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$, the skeleton decomposition is defined by:
\begin{equation}
\mathbf{A} \simeq \mathbf{A}(\cdot, \mathcal{J}) \mathbf{A}(\mathcal{I}, \cdot)^{-1} \mathbf{A}(\mathcal{I}, \cdot) ,
\end{equation}
where $\mathcal{I} = (i_1, \ldots, i_r)$ and $\mathcal{J} = (j_1, \ldots, j_r)$ are subsets of the index sets $[1, \ldots, m]$ and $[1, \ldots, n]$. The selection of the indices $\{\mathcal{I}, \mathcal{J}\}$ need to be such that most of the information contained in $\mathbf{A}$ is captured by the decomposition. It turns out that the optimal submatrix $\mathbf{A}(\mathcal{I}, \mathcal{J})$ is that with maximal determinant in modulus among all the $r \times r$ submatrices of $\mathbf{A}$ \cite{21}. The problem of finding such a matrix is NP-hard \cite{8}. An approximation to the solution of this problem can be found using the maxvol algorithm \cite{21}, in a row-column alternating fashion as explained in \cite{16}. Running maxvol is computationally inexpensive and requires $2c(n-r)r$ operations, where $c$ is usually a small constant in many practical applications.

The problem of finding the TT-decomposition $\mathbf{A}_{TT}$ can be cast as the minimization problem
\begin{equation}
\min_{G_1, \ldots, G_d} \|\mathbf{A} - \mathbf{A}_{TT}\|_F .
\end{equation}
One possible approach for solving this problem is TT-cross \cite{16}. Here the optimization is performed through left-to-right and right-to-left sweeps of the cores, using the matrix skeleton decomposition.
to find the most relevant fibers in the $d$ dimensional space. A fiber is, for a $d$-dimensional tensor $\mathbf{A}$, the equivalent of what rows and columns are for a matrix. In MATLAB notation, the $(i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_d)$ fiber along the $k$-th dimension is $\mathbf{A}(i_1, \ldots, i_{k-1}; i_{k+1}, \ldots, i_d)$. This approach provides linear scaling in the number of entries evaluated. On the other hand, it requires the TT-ranks to be known a priori in order to select the correct number of fibers for each dimension. Underestimating these ranks leads to a poor (and in some cases erroneous) approximation, while overestimating them increases computational effort.

A more effective approach is the TT-DMRG-cross \[48\], where the optimization is performed over two cores, $G_k$ and $G_{k+1}$, at a time. At step $k$ of the sweeps, the core $W_k(i_k, i_{k+1}) = G_k(i_k)G_k(i_{k+1})$ solving (2.10) is found, and the cores $G_k$ and $G_{k+1}$ are recovered through the SVD of $W_k$. The relevant core $W_k$ is identified again using the maximum volume principle, by selecting the most important planes $\mathbf{A}(i_1, \ldots, i_{k-1}; i_{k+1}, \ldots, i_d)$ in the $d$-dimensional space. Unlike TT-cross, this method is rank-revealing, meaning that the TT-ranks do not need to be guessed a priori; instead, the method determines them automatically.

3. Relevant results from approximation theory. The main objective of this work is to extend the TT format to functional approximations of $f$. To do this we need to consider the case where some smoothness can be assumed on $f$. Here we will review some concepts from polynomial approximation theory which, in subsequent sections, will be combined with the tensor-train decomposition. In the following, we will make use of the Sobolev spaces:

\begin{equation}
(3.1) \quad \mathcal{H}_{\mu}^k(I) = \left\{ f \in L^2_{\mu}(I) : \sum_{|i| \leq k} \|D^{(i)}f\|_{L^2_{\mu}(I)} < +\infty \right\},
\end{equation}

where $k \geq 0$, $D^{(i)}f$ is the $i$-th weak derivative of $f$, $I = I_1 \times \cdots \times I_d$ is a product of intervals of $\mathbb{R}$ and $\mu : \mathcal{B}(I) \to \mathbb{R}$ is a $\sigma$-finite measure on the Borel $\sigma$-algebra defined on $I$. This space is equipped with the norm $\| \cdot \|_{\mathcal{H}_{\mu}^k(I)}^2$ defined as

\begin{equation}
(3.2) \quad \|f\|_{\mathcal{H}_{\mu}^k(I)}^2 = \sum_{|i| \leq k} \|D^{(i)}f\|^2_{L^2_{\mu}(I)}
\end{equation}

and the semi-norm $| \cdot |_{1,\mu,k}$ given by

\begin{equation}
(3.3) \quad |f|_{1,\mu,k}^2 = \sum_{|i| = k} \|D^{(i)}f\|^2_{L^2_{\mu}(I)}.
\end{equation}

In the following we will assume that $\mu$ is a product measure satisfying $\mu(I) = \prod_{i=1}^d \mu_i(I_i)$, where $\mu_i$ is a $\sigma$-finite measure on the Borel $\sigma$-algebra defined on $I_i$.

3.1. Projection. A function $f \in L^2_{\mu}(I)$ can be approximated by its projection onto a finite-dimensional subspace of $L^2_{\mu}(I)$. The following results hold both for compact and non-compact supports $I$.

**Definition 3.1.** (Spectral expansion). Let $I \subseteq \mathbb{R}^d$ and $f \in L^2_{\mu}(I)$. Let $\{\Phi_i\}_{i=0}^{\infty}$ be a set of multivariate polynomials forming an orthonormal basis for $L^2_{\mu}(I)$, where $\Phi_i(x) = \phi_{i_1,1}(x_1) \cdots \phi_{i_d,d}(x_d)$, $i = (i_1, \ldots, i_d)$, and $\phi_{i,j}$ is the degree-$i$ member of the family of univariate polynomials orthonormal with respect to the measure $\mu_j$. For $\mathbf{N} = (N_1, \ldots, N_d) \in \mathbb{N}_0^d$, the degree-$\mathbf{N}$ spectral expansion of $f$ is obtained from the projection operator $P_{\mathbf{N}} : L^2_{\mu}(I) \to \text{span}(\{\Phi_i\}_{i=0}^{\mathbf{N}})$, where

\begin{equation}
(3.4) \quad P_{\mathbf{N}}f = \sum_{0 \leq i \leq \mathbf{N}} c_i\Phi_i, \quad c_i = \int_I f \Phi_id\mu(x).
\end{equation}

and $i \leq \mathbf{N}$ denotes $\bigwedge_{1 \leq j \leq d} (i_j \leq N_j)$. 

For simplicity, in the following we define $P_N := P_N$ when $N_1 = \ldots = N_d = N$. The rate of convergence of the spectral expansion (3.4) is determined by the smoothness of $f$.

**Proposition 3.2** (Convergence of spectral expansion [29, 6]). Let $f \in H^k_\mu(I)$ for $k \geq 0$. Then

$$
\|f - P_N f\|_{L^2_\mu(I)} \leq C(k) N^{-k} \|f\|_{L^\infty_\mu}. 
$$

In practice the coefficients $c_i$ in (3.4) are approximated using discrete inner products based on quadrature rules of sufficient accuracy. We will focus here on $d$-dimensional quadrature rules produced by tensorizing univariate Gaussian rules—specifically, for dimension $i$, an $(N_i + 1)$-point Gaussian quadrature rule [10]. Let $(x_i, w_i)_{i=0}^N$ be the points and weights describing such a rule [20].

A $d$-dimensional integral can then be approximated by:

$$
\int_I f(x) \, d\mu(x) \approx \sum_{i=0}^N f(x_i) w_i =: U_N(f).
$$

The discrete (and computable) version of the spectral expansion (3.4) is then defined as follows.

**Definition 3.3** (Discrete projection). Let $(x_i, w_i)_{i=0}^N$ be a set of quadrature points and weights. The discrete projection of $f$ is obtained by the action of the operator $\hat{P}_N : L^2_\mu(I) \to \text{span}(\{\Phi_i\}_{i=0}^N)$, defined as

$$
\hat{P}_N f = \sum_{i=0}^N \hat{c}_i \Phi_i, \quad \hat{c}_i = U_N(f \Phi_i) = \sum_{i=0}^N f(x_i) \Phi_i(x_i) w_i.
$$

This approximation to the orthogonal projection onto $\mathbb{P}_N$, the space of polynomials of degree up to $N$, is sometimes called a pseudospectral approximation. For simplicity, we have focused on the fully tensorized case and assumed a certain level of smoothness in between them. Here we will consider piecewise linear interpolation and polynomial interpolation on closed and bounded domains $I = I_1 \times \cdots \times I_d$. Other interpolation rules could be used inside the same framework for specific problems.

**3.2. Interpolation.** A function $f$ can also be approximated using interpolation on a set of nodes and assuming a certain level of smoothness in between them. Here we will consider piecewise linear interpolation and polynomial interpolation on closed and bounded domains $I = I_1 \times \cdots \times I_d$. Other interpolation rules could be used inside the same framework for specific problems.

The linear interpolation of a function $f : [a, b] \to \mathbb{R}$ can be written in terms of basis functions called hat functions: given a set of distinct ordered nodes $\{x_i\}_{i=0}^N \in [a, b]$ with $x_0 = a$ and $x_N = b$, the hat functions are:

$$
e_i(x) = \begin{cases} 
\frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if } x_{i-1} \leq x \leq x_i \land x \geq a \\
\frac{x - x_{i+1}}{x_i - x_{i+1}} & \text{if } x_i \leq x \leq x_{i+1} \land x \leq b \\
0 & \text{otherwise}
\end{cases}.
$$

When dealing with multiple dimensions, several options are available. A very common choice of basis functions have their support over simplices around a node. This allows the basis functions $e_i$ to remain linear. In this paper, we will instead use basis functions supported on the hypercubes adjacent to a node. These basis functions $e_i$ can no longer be linear while preserving the linear interpolation property: they need to be bilinear in two dimensions, trilinear in three dimensions, and so on. Letting $V$ be the set of piecewise continuous functions on $I$, the multi-linear interpolation operator $I_N : V \to C^0(I)$ is then defined by

$$
I_N f(x) = \sum_{i=0}^N \hat{c}_i e_i(x), \quad \hat{c}_i = f(x_i),
$$
where \( \{ x_i \}_{i=0}^N \) is a tensor grid of points. Again we will use the notation \( I_N := I_N^N \) when \( N_1 = \ldots = N_d = N \). If the grid points are uniformly distributed, the convergence of this approximation is as follows.

**Proposition 3.4 (Convergence of linear interpolation [4]).** Let \( f \in H^2(\Omega) \). Then

\[
\| f - I_N f \|_{L^2(\Omega)} \leq CN^{-2} |f|_{1, \mu, 2}.
\]

The second type of interpolation we will use in this paper is Lagrange polynomial interpolation. It is based on the Lagrange polynomials \( \{ l_i \}_{i=1}^N \), defined in the univariate case as

\[
l_i(x) = \prod_{0 \leq m < k \atop m \neq i} \frac{x - x_m}{x_i - x_m},
\]

where the nodes \( \{ x_i \}_{i=1}^N \in [a, b] \) are typically distributed non-uniformly over the interval; an example is the Gauss nodes used in Section 3.1. This choice is designed to avoid the Runge phenomenon and hence assure a more accurate approximation. The univariate polynomial interpolation operator \( \Pi_N : V \to \text{span} \{ l_i \}_{i=1}^N \) is given by

\[
\Pi_N f(x) = \sum_{i=0}^N \hat{c}_i l_i(x), \quad \hat{c}_i = f(x_i).
\]

Lagrange interpolation in the multivariate case presents many theoretical issues when used for interpolation on arbitrary nodes. In the scope of this paper, however, we will only consider tensor grids of nodes, for which the theory follows easily from the univariate case. As we will see in the next section, we will never explicitly construct these tensor grids, thanks to the TT decomposition and cross-interpolation. But the convergence properties of Lagrange interpolation on tensor grids will nonetheless be useful for analysis purposes. The convergence of the Lagrange interpolant is again dictated by the smoothness of the function being approximated.

**Proposition 3.5 (Convergence of Lagrange interpolation [2, 6]).** Let \( f \in H^k(\Omega) \) for \( k \geq 1 \). Then

\[
\| f - \Pi_N f \|_{L^2(\Omega)} \leq C(k)N^{-k} |f|_{1, \mu, k}.
\]

Recall that Lagrange interpolation on \( N + 1 \) Gauss nodes is equivalent to the degree-\( N \) pseudospectral approximation (discrete projection) computed with the same nodes [3]; this equivalence also extends to the tensorized case.

### 4. Spectral tensor-train decomposition

Now we blend the discrete tensor-train decomposition of Section 2.2 with the polynomial approximations described in Section 3. First, we construct a continuous version of the tensor-train decomposition, termed the functional tensor-train (FTT) decomposition. The construction proceeds by recursively decomposing non-symmetric square integrable kernels through auxiliary symmetric square integrable kernels, as in [49]. Next, we prove that this decomposition converges under certain regularity conditions, and that the cores of the FTT decomposition inherit the regularity of the original function, and thus are amenable to spectral approximation when the original function is smooth. Based on this analysis, we propose an efficient approach to high-dimensional function approximation that employs only one-dimensional polynomial approximations of the cores of the FTT decomposition, and we analyze the convergence of these approximations.
4.1. Functional tensor-train decomposition. Let $X \times Y$ and let $f$ be a Hilbert-Schmidt kernel with respect to the finite measure $\mu : B(X \times Y) \to \mathbb{R}$, i.e., $f \in L^2_{\mu}(X \times Y)$. We restrict our attention to product measures, so $\mu = \mu_x \times \mu_y$. The operator

$$T : L^2_{\mu_y}(Y) \to L^2_{\mu_x}(X)$$

$$g \mapsto \int_Y f(x, y) g(y) d\mu_y(y)$$

is linear, bounded and compact [25 Cor. 4.6]. The Hilbert adjoint operator of $T$ is $T^* : L^2_{\mu_x}(X) \to L^2_{\mu_y}(Y)$. Then $T T^* : L^2_{\mu_x}(X) \to L^2_{\mu_y}(Y)$ is a compact Hermitian operator. By the spectral theory of compact operators, the spectrum of $T T^*$ comprises a countable set of eigenvalues and the only point of accumulation is zero [36 Thm 8.3-1, 8.6-4]. Since $T T^*$ is self-adjoint, its eigenfunctions $\{\gamma(x; i)\}_{i=1}^{\infty} \subset L^2_{\mu_x}(X)$ form an orthonormal basis [25 Cor. 4.7]. The operator $TT^* : L^2_{\mu_y}(Y) \to L^2_{\mu_y}(Y)$ is also self-adjoint and compact, with eigenfunctions $\{\varphi(i; y)\}_{i=1}^{\infty} \subset L^2_{\mu_y}(Y)$ and the same eigenvalues as $T T^*$. Then we have the following expansion of $f$.

**DEFINITION 4.1 (Functional SVD).** Let the integral operators $T T^*$ and $T^* T$ have eigenvalues $\{\lambda(i)\}_{i=1}^{\infty}$ and associated eigenfunctions $\{\gamma(x; i)\}_{i=1}^{\infty}$ and $\{\varphi(i; y)\}_{i=1}^{\infty}$, respectively. Then the functional SVD of $f$ is:

$$f = \sum_{i=1}^{\infty} \sqrt{\lambda(i)} \gamma(\cdot; i) \otimes \varphi(i; \cdot).$$

In the general setting considered here, the convergence of (4.2) is in $L^2_{\mu}$. Now let $I_1 \times \cdots \times I_d = I \subseteq \mathbb{R}^d$ and let $f$ be a Hilbert-Schmidt kernel with respect to the finite measure $\mu : B(I) \to \mathbb{R}$, i.e., $f \in L^2_{\mu}(I)$. We assume $\mu = \prod_{i=1}^{d} \mu_i$. Applying the functional SVD to $f$ with $X = I_1$ and $Y = I_2 \times \cdots \times I_d$, we obtain

$$f(x) = \sum_{\alpha_1=1}^{\infty} \sqrt{\lambda_1(\alpha_1)} \gamma_1(x; \alpha_1) \varphi_1(\alpha_1; x_2, \ldots, x_d).$$

Now let $X = N \times I_2$, $Y = I_3 \times \cdots \times I_d$, and $\tau$ be the counting measure on $N$. From the definition of counting measure, the orthonormality of $\varphi(\alpha_i; \cdot)$ for all $\alpha_i \in N$, and the fact that $f \in L^2_{\mu}(I)$, we have:

$$\int_{X \times Y} \left| \sqrt{\lambda_1(\alpha_1)} \varphi_1(\alpha_1; x_2, \ldots, x_d) \right|^2 d\tau(\alpha_1) d\mu_2(x_2) \cdots d\mu_d(x_d) = \sum_{\alpha_1=1}^{\infty} \lambda_1(\alpha_1) < \infty.$$

This means that $(\sqrt{\lambda_1(\alpha_1)} \varphi_1(\alpha_1; x_2, \ldots, x_d)) \in L^2_{\mu_2 \times \cdots \times \mu_d}(X \times Y)$ and thus it is a Hilbert-Schmidt kernel. Then, using the functional SVD we obtain

$$\sqrt{\lambda_1(\alpha_1)} \varphi_1(\alpha_1; x_2, \ldots, x_d) = \sum_{\alpha_2=1}^{\infty} \sqrt{\lambda_2(\alpha_2)} \gamma_2(\alpha_2; x_2; \alpha_2) \varphi_2(\alpha_2; x_3, \ldots, x_d).$$

This expansion can now be plugged into (4.3):

$$f(x) = \sum_{\alpha_1=1}^{\infty} \sum_{\alpha_2=1}^{\infty} \sqrt{\lambda_2(\alpha_2)} \gamma_1(x; \alpha_1) \gamma_2(\alpha_1; x_2; \alpha_2) \varphi_2(\alpha_2; x_3, \ldots, x_d).$$

**SPECTRAL TENSOR-TRAIN DECOMPOSITION**
where $\alpha \in \mathbb{R}^d$ and $\gamma_d(a_{d-1};x_d,a_d) = \sqrt{\lambda_{d-1}(a_{d-1})}\phi_d(a_{d-1};x_d)$. We will call this format the functional tensor-train (FTT) decomposition.

If we now truncate the FTT decomposition, we obtain the functional version of the tensor-train approximation.

**Definition 4.2 (FTT approximation).** Let $I_1 \times \cdots \times I_d = I \subseteq \mathbb{R}^d$ and $f \in L^2_{\mu}(I)$. For $r = (1, r_1, \ldots, r_{d-1}, 1)$, a functional $TT$-rank-$r$ approximation of $f$ is:

\begin{equation}
\begin{aligned}
f_{TT}(x) := \sum_{\alpha_0, \ldots, \alpha_d = 1}^r \gamma_1(\alpha_0, x_1, \alpha_1) \cdots \gamma_d(\alpha_{d-1}, x_d, \alpha_d),
\end{aligned}
\end{equation}

where $\gamma_i(\alpha_{i-1}, \cdot, \alpha_i) \in L^2_{\mu_i}$ and $(\gamma_k(i, \cdot, m), \gamma_k(i, \cdot, n))_{L^2_{\mu_k}} = \delta_{mn}$. The residual of this approximation will be denoted by $R_{TT} := f - f_{TT}$. We will call $\{\gamma_i\}_{i=1}^d$ the cores of the approximation.

**Proposition 4.3.** Let the functional tensor-train decomposition (4.7) be truncated retaining the largest singular values $\{(\sqrt{\lambda_i(\alpha_i)})_{\alpha_i = 1}^d\}_{i=1}$. Then the approximation (4.8) fulfills the condition:

\begin{equation}
\begin{aligned}
\|R_{TT}\|_{L^2_{\mu}}^2 = \min_{g \in L^2_{\mu}} \|f - g\|_{L^2_{\mu}}^2 = \sum_{i=1}^{d-1} \left( \prod_{j=1}^{i-1} r_j \right) \sum_{\alpha_i = r_i+1}^\infty \lambda_i(\alpha_i).
\end{aligned}
\end{equation}

**Proof.** The first equality is due to the construction of $f_{TT}$ by a sequence of orthogonal projections that minimize the error in the $L^2_{\mu}$-norm. These projections are onto the subspaces spanned by the eigenfunctions of the Hermitian operators induced by the tensor $f$, and are thus optimal \[53, 59\].

The error bound is obtained by induction. The first step of the decomposition (4.3) leads to:

\begin{equation}
\begin{aligned}
\|f - f_{TT}\|_{L^2_{\mu}}^2 = \left\| f - \sum_{\alpha_1 = 1}^{r_1} \sqrt{\lambda_1} \gamma_1 \varphi_1 + \sum_{\alpha_1 = 1}^{r_1} \sqrt{\lambda_1} \gamma_1 \varphi_1 - f_{TT} \right\|_{L^2_{\mu}}^2,
\end{aligned}
\end{equation}

where the Pythagorean theorem was used in light of the fact that

\begin{equation}
\begin{aligned}
\left\langle f - \sum_{\alpha_1 = 1}^{r_1} \sqrt{\lambda_1} \gamma_1 \varphi_1, \sum_{\alpha_1 = 1}^{r_1} \sqrt{\lambda_1} \gamma_1 \varphi_1 - f_{TT} \right\rangle_{L^2_{\mu}} = 0,
\end{aligned}
\end{equation}

the orthonormality of $\{\gamma_1(\alpha_1; \cdot)\}$ and $\{\varphi_1(\alpha_1; \cdot)\}$ was exploited and the arguments of $\lambda_1$, $\gamma_1$ and $\varphi_1$ were omitted in order to simplify the notation. Next, let $(\sqrt{\lambda_1} \varphi_1)(\alpha_1; x_2, \ldots, x_d) :=
SPECTRAL TENSOR-TRAIN DECOMPOSITION

\[ \sqrt{\lambda_1(\alpha_1)} \varphi_1(\alpha_1; x_2, \ldots, x_d) \] and apply the second step of the decomposition to the last term of (4.10):

\[
\left\| \sum_{\alpha_1=1}^{r_1} \gamma_1 \left( \sqrt{\lambda_1} \varphi_1 \right) - f_{TT} \right\|_{L_2}^2 = \left\| \sum_{\alpha_1=1}^{r_1} \gamma_1 \left( \sqrt{\lambda_1} \varphi_1 \right) - \sum_{\alpha_1=1}^{r_1} \sum_{\alpha_2=1}^{r_2} \gamma_1 \sqrt{\lambda_2 \gamma_2} \varphi_2 \right\|_{L_2}^2
\]

Plugging (4.12) into (4.10) and proceeding by induction, the bound (4.9) is obtained.

The result given in Proposition 4.3 does not directly involve any properties of the function \( f \). Now we will try to link the error of the FTT approximation with the regularity of \( f \). To do so, we will use the following auxiliary results: Proposition 4.4, which is a particular case of [50, Prop. 2.21], and Lemmas 4.5 and 4.6 which are proven in Appendix [B].

**Proposition 4.4.** Let \( I = [0, 1]^d \) be a bounded domain and \( V \in L^2_{\mu}(I \times I) \) be the symmetric kernel of the compact non-negative integral operator \( \mathcal{V} : L^2_{\mu}(I) \to L^2_{\mu}(I) \). If \( V \) is \( \mathcal{H}^k_{\mu}(I \times I) \) with \( k > 0 \) and \( \{\lambda_m\}_{m \geq 1} \) denotes the eigenvalue sequence of \( V \), then

\[
\lambda_m \leq |V|_{L^2_{\mu}(I \times I)} m^{-k/d} \quad \forall m \geq 1.
\]

**Lemma 4.5.** Let \( f \in \mathcal{H}^k_{\mu}(I) \), \( \bar{I} = I_2 \times \ldots \times I_d \) and \( J(x,\bar{x}) = \langle f(x,y), f(\bar{x},y) \rangle_{L^2_{\mu}(I)} \). Then, \( J \in \mathcal{H}^k_{\mu}(I_1 \times I) \) and

\[
|J|_{L^1_{\mu}(I_1 \times I)} \leq \|f\|^2_{\mathcal{H}^k_{\mu}(I)}.
\]

**Lemma 4.6.** Let \( \sqrt{\lambda_i} \varphi_i \) \((\alpha_i; \cdot) \in \mathcal{H}^k_{\mu}(I)\), \( \bar{I} = I_{1+1} \times \ldots \times I_d \), \( \bar{I} = I_{1+2} \times \ldots \times I_d \), \( \sqrt{\lambda_0} \varphi_0 := f \) and let

\[
\left( \sqrt{\lambda_i} \varphi_i \right)_{TT} (\alpha_i; \cdot) := \sum_{\alpha_{i+1}=1}^{r_{i+1}} \sqrt{\lambda_{i+1}(\alpha_{i+1})} \gamma_{i+1} (\alpha_{i+1}; x_{i+1}; \alpha_{i+1}) \varphi_{i+1} (\alpha_{i+1}; x_{i+2}, \ldots, x_d)
\]

be the functional SVD approximation of \( \sqrt{\lambda_i} \varphi_i \) \((\alpha_i; \cdot) \). Then, for all \( \alpha_i \in [1, \ldots, r_i] \)

\[
\| \varphi_{i+1} (\alpha_{i+1}) \|^2_{\mathcal{H}^k_{\mu}(I)} \leq \frac{1}{\lambda_{i+1}(\alpha_{i+1})} \left\| \left( \sqrt{\lambda_i} \varphi_i \right) (\alpha_i) \right\|^2_{\mathcal{H}^k_{\mu}(I)} \leq \frac{1}{\lambda_{i+1}(\alpha_{i+1})} \|f\|^2_{\mathcal{H}^k_{\mu}(I)}.
\]

For the sake of simplicity, in the following analysis we will let the ranks be \( r = (r, \ldots, r) \). Our main result, relating the regularity of \( f \), the ranks \( r \), and the input dimension \( d \) to the error of the FTT approximation, is as follows.

**Theorem 4.7 (FTT-approximation convergence).** Let \( f \in \mathcal{H}^k_{\mu}(I) \), then

\[
\|R_{TT}\|^2_{L_2} \leq \|f\|^2_{\mathcal{H}^k_{\mu}(I)} \zeta(k, r+1) \frac{r^d - r}{r(r-1)} \quad \text{for } r > 1,
\]
where ζ is the Hurwitz Zeta function. Furthermore

\[
\lim_{r \to \infty} \| R_{TT} \|_{L^2(R)}^2 \leq \| f \|_{H^2(I)}^2 \frac{1}{(d-2)} \text{ for } k = d - 1
\]

and

\[
\lim_{r \to \infty} \| R_{TT} \|_{L^2(R)}^2 = 0 \quad \text{for } k > d - 1.
\]

**Proof.** We start by considering the case \( I = I_1 \times I_2 \times I_3 \) and we define the following approximations of \( f \), using the functional SVD (4.2):

\[
f_{TT,1} = \sum_{\alpha_1=1}^{r_1} \sqrt{\lambda_1(\alpha_1)} \gamma_1(x_1; \alpha_1) \varphi_1(\alpha_1; x_2, x_3),
\]

\[
f_{TT} = \sum_{\alpha_1=1}^{r_2} \gamma_1(x_1; \alpha_1) \left( \sqrt{\lambda_1(\varphi_1)} \right)_{TT,1} (\alpha_1; x_2, x_3),
\]

where

\[
\left( \sqrt{\lambda_1(\varphi_1)} \right)_{TT,1} (\alpha_1; x_2, x_3) = \sum_{\alpha_2=1}^{r_2} \sqrt{\lambda(\alpha_2)} \gamma_2(\alpha_2; x_2; \alpha_2) \varphi_2(\alpha_2; x_3).
\]

As for (4.11), it is possible to show that \( \langle f - f_{TT,1}, f_{TT,1} - f_{TT} \rangle_{L^2(I)} = 0 \) and so

\[
\| R_{TT} \|_{L^2(I)}^2 = \| f - f_{TT} \|_{L^2(I)}^2 = \| f - f_{TT,1} \|_{L^2(I)}^2 + \| f_{TT,1} - f_{TT} \|_{L^2(I)}^2.
\]

Exploiting the orthogonality of the singular functions, Proposition 4.4, and Lemma 4.5 we have

\[
\| f - f_{TT,1} \|_{L^2(I)}^2 = \sum_{\alpha_1=1}^{\infty} \lambda(\alpha_1) \leq \sum_{\alpha_2=1}^{\infty} \alpha_2^{-k} |J_0(1)| \zeta(k, r_1 + 1),
\]

where \( J_0(x_1, \bar{x}_1) = \langle f(x_1, x_2, x_3), f(\bar{x}_1, x_2, x_3) \rangle_{L^2(I_2 \times I_3)} \). Similarly:

\[
\| \left( \sqrt{\lambda_1(\varphi_1)} \right)_{TT,1} (\alpha_1) - \left( \sqrt{\lambda_1(\varphi_1)} \right)_{TT} (\alpha_1) \|_{L^2(I_2 \times I_3)}^2 \leq \sum_{\alpha_2=2}^{\infty} \alpha_2^{-k} |J_1(\alpha_1)| \zeta(k, r_2 + 1),
\]

where \( J_1(\alpha_1; x_2, \bar{x}_2) = \langle \left( \sqrt{\lambda_1(\varphi_1)} \right) (\alpha_1; x_2, x_3), \left( \sqrt{\lambda_1(\varphi_1)} \right) (\alpha_1; x_2, \bar{x}_2) \rangle_{L^2(I_3)} \). With the help of Lemma 4.6 this leads to

\[
\| f_{TT,1} - f_{TT} \|_{L^2(I)}^2 = \sum_{\alpha_1=1}^{r_1} \gamma_1(\alpha_1) \| \left( \sqrt{\lambda_1(\varphi_1)} \right)_{TT,1} (\alpha_1) - \left( \sqrt{\lambda_1(\varphi_1)} \right)_{TT} (\alpha_1) \|_{L^2(I)}^2
\]

\[
= \sum_{\alpha_1=1}^{r_1} \gamma_1(\alpha_1) \| \left( \sqrt{\lambda_1(\varphi_1)} \right)_{TT} (\alpha_1) \|_{L^2(I_2 \times I_3)}^2 \zeta(k, r_2 + 1) \leq r_1 \| f \|_{H^2(I)}^2 \zeta(k, r_2 + 1).
\]
Thus we obtain the bound
\begin{equation}
\|R_{TT}\|_{L^2_{µ}^{2}(I)}^{2} \leq \|f\|_{H^{2}_{µ}(I)}^{2} \left[\zeta(k,r_{1}+1) + r_{1}\zeta(k,r_{2}+1)\right].
\end{equation}

Now let \( I = I_{1} \times \ldots \times I_{d} \) and \( r = (r_{1}, \ldots, r_{d}) \), for \( r \geq 2 \); then
\begin{equation}
\|R_{TT}\|_{L^2_{µ}^{2}(I)}^{2} \leq \|f\|_{H^{2}_{µ}(I)}^{2} \sum_{i=1}^{d-1} \left(\prod_{j=1}^{i-1} r_{j}\right) \zeta(k,r_{i}+1)
\end{equation}
\begin{equation}
= \|f\|_{H^{2}_{µ}(I)}^{2} \zeta(k,r_{1}+1) \sum_{i=1}^{d-1} r^{i-1} = \|f\|_{H^{2}_{µ}(I)}^{2} \zeta(k,r_{1}+1) \frac{r^{d}-r}{r(r-1)},
\end{equation}
where one can recognize the form [4.9] of the error. The condition \( r \geq 2 \) was introduced only in order to obtain the last equality, which is useful for the following asymptotic analysis and proves the first part of the theorem. In general only \( r \geq 1 \) is required.

Let us now study the asymptotic behavior of \( \|R_{TT}\|_{L^2_{µ}^{2}(I)}^{2} \) as \( r \to \infty \). For \( k > 1 \), we can use the bound:
\begin{equation}
\zeta(k,r_{1}+1) = \sum_{i=r+1}^{\infty} i^{-k} \leq \int_{r+1}^{\infty} i^{-k} di = \frac{(r+1)^{-(k-1)}}{(k-1)}.
\end{equation}

Plugging this into (4.28) and considering its asymptotic behavior as \( r \to \infty \), we obtain:
\begin{equation}
\|R_{TT}\|_{L^2_{µ}^{2}(I)}^{2} \leq \|f\|_{H^{2}_{µ}(I)}^{2} \left(\frac{(r+1)^{-(k-1)}}{(k-1)} \frac{r^{d}-r}{r(r-1)}\right) \approx \|f\|_{H^{2}_{µ}(I)}^{2} \frac{1}{(k-1)r_{k-1}} \frac{r^{d-1-k}}{r} = \|f\|_{H^{2}_{µ}(I)}^{2} \frac{r^{d-1-k}}{k-1}.
\end{equation}

This leads to the two asymptotic estimates (4.18) and (4.19), completing the proof.

4.2. Regularity of the FTT decomposition. In order to apply the traditional polynomial approximation theory to the functional tensor-train decomposition, we need that such decomposition retains the same regularity of the original function. In particular, in the scope of the polynomial approximation theory presented in Section 3, we need the boundedness of the weak derivatives used for the definition of Sobolev spaces (3.1). With this perspective, we will need the absolute convergence almost everywhere of the FTT decomposition. Smithies [53, Thm. 14] proved that a kind of integrated Hölder continuity with exponent \( \alpha > 1/2 \) is a sufficient condition for the absolute convergence almost everywhere (a.e.) of the functional SVD. The condition required by Smithies is a generalization of the Hölder continuity a.e. [56] as we show in Appendix A. The Smithies’ result can be extended by construction to the FTT decomposition:

**Corollary 4.8 (Absolute convergence almost everywhere).** Let \( I_{1} \times \ldots \times I_{d} = I \subset \mathbb{R}^{d} \) be closed and bounded, and \( f \in L^{2}_{µ}(I) \) be a Hölder continuous function with exponent \( \alpha > 1/2 \). Then the FTT decomposition (4.7) converges absolutely almost everywhere.

Now we can prove that if \( f \) belongs to a certain Sobolev space, then also the cores of the FTT decomposition will belong to the same Sobolev space.

**Theorem 4.9 (FTT decomposition and Sobolev spaces).** Let \( I_{1} \times \ldots \times I_{d} = I \subset \mathbb{R}^{d} \) be closed and bounded, and \( f \in L^{2}_{µ}(I) \) be a Hölder continuous function with exponent \( \alpha > 1/2 \) such that \( f \in H^{k}_{µ}(I) \). Then the FTT decomposition (4.7) is such that \( \gamma_{j}(\alpha_{j-1}, \ldots, \alpha_{j}) \in H^{k}_{µ}(I_{j}) \) for all \( j \), \( \alpha_{j-1} \) and \( \alpha_{j} \).

**Proof.** We will first show this property for the functional SVD (4.2) of the Hölder \( \alpha > 1/2 \) continuous function \( f \in H^{k}_{µ}(X \times Y) \). First we want to show that
\begin{equation}
D^{i}f = \sum_{j=1}^{\infty} \sqrt{\lambda_{j}} (D^{i_{1}}\psi_{j} \otimes D^{i_{2}}\phi_{j}),
\end{equation}
where \( i = (i_1, i_2) \). Since \( f \) is Hölder \((\alpha > 1/2)\) continuous, (4.2) converges absolutely a.e. by Smithies [53], then we can define

\[
\inf > g := \sum_{j=1}^{\infty} \sqrt{\lambda_j} (\psi_j \otimes \phi_j) \geq \sum_{j=1}^{\infty} \sqrt{\lambda_j} (\psi_j \otimes \phi_j),
\]

where the domination holds almost everywhere. The series (4.2) is convergent almost everywhere by Smithies [53]. By the definition of weak derivative, for all \( D \)
\[
\text{Using the Cauchy-Schwarz inequality:}
\]
\[
\text{where the last bound is due to the fact that} \ D \ \text{have that}
\]
\[
\text{have that}
\]
\[
\text{Thus this holds also for any} \ \chi = \chi_x \otimes \chi_y \in C_c^\infty(X \times Y). \text{ Using the dominated convergence theorem, we obtain:}
\]
\[
(-1)^i \int_{X \times Y} D^i f \chi d\mu = \int_{X \times Y} f \chi^{(i)} d\mu.
\]
\[
\text{Thus (4.31) holds. Next we want to show that} \ f \in \mathcal{H}_\mu^k(X \times Y) \implies \|D^i \psi_j\|_{L^2_\mu(X)} < \infty \text{ and}
\]
\[
\|D^{i_1} \phi_j\|_{L^2_\mu(Y)} < \infty \text{ for} \ i_1, i_2 \leq k. \text{ Thanks to (4.31) and due to the orthonormality of} \ \{\phi_j\}_{j=1}^{\infty} \text{ we have that}
\]
\[
D^{i_1} \psi_j = \frac{1}{\sqrt{\lambda_j}} \left\langle D^{(i_1,0)} f, \phi_j \right\rangle_{L^2_\mu(Y)}.
\]
\[
\text{Using the Cauchy-Schwarz inequality:}
\]
\[
\|D^{i_1} \psi_j\|_{L^2_\mu(X)}^2 = \left\langle D^{(i_1,0)} f, \phi_j \right\rangle_{L^2_\mu(Y)} \|D^{(i_1,0)} f\|_{L^2_\mu(X)}^2
\]
\[
\leq \frac{1}{\lambda_j} \left\| \phi_j \right\|^2_{L^2_\mu(Y)} \|D^{(i_1,0)} f\|^2_{L^2_\mu(X \times Y)} < \infty,
\]

where the last bound is due to the fact that \( \{\phi_j\}_{j=1}^{\infty} \subset L^2_\mu(Y) \) — see Eqs. (4.1) and (4.3) — and \( D^{(i_1,0)} f \in L^2_\mu(X \times Y) \) because \( i_1 \leq k \) and \( f \in \mathcal{H}_\mu^k(X \times Y) \). In the same way \( \|D^{i_2} \phi_j\|_{L^2_\mu(Y)} < \infty \) for all \( i_2 \leq k \). It follows that \( \{\psi_j\}_{j=1}^{\infty} \subset \mathcal{H}_\mu^k(X) \) and \( \{\phi_j\}_{j=1}^{\infty} \subset \mathcal{H}_\mu^k(Y) \). The extension to the FTT decomposition (4.7) follows by its construction in terms of repeated functional SVDs.

**Remark 1.** The results above have the limitation of holding for functions defined on closed and bounded domains. In many practical cases, however, functions are defined on the real line, equipped with a finite measure. To the author’s knowledge, the corresponding result for such cases has not been proven in literature. The result by Smithies [53] Thm. 14] hinges on a result by Hardy and Littlewood [27] Thm. 10] on the convergence of Fourier series, and this is the only passage in the proof where the closedness and boundedness of the domain is explicitly used. A similar result for an orthogonal system in \( L^2_\mu(-\infty, \infty) \), where \( \mu \) is a finite measure, would be sufficient to extend
Smithies’ result to the real line. For one of the numerical examples presented in the following (Sec. 5.3), we will assume that this result holds.

Other regularity properties can be proven, given different kinds of continuity of the function \( f \). These properties are not strictly necessary in the scope of polynomial approximation theory, so we will state them without proof. The first regards the continuity of the cores of the FTT decomposition and follows directly from Mercer’s theorem [30].

**Proposition 4.10** (Continuity). Let \( I_1 \times \cdots \times I_d = I \subset \mathbb{R}^d \), and \( f \in L^2_\mu(I) \) be a continuous function with FTT decomposition \((4.7)\). Then \( \gamma_i(\alpha_{i-1}, \cdot, \alpha_i) \) are continuous for every \( i \) and \( \alpha_i \).

The second property regards the strong derivatives of the cores of the FTT decomposition. It requires the Lipschitz continuity of the function and then follows from a result on the uniform convergence of the functional SVD by Hammerstein [26, 56].

**Theorem 4.11** (Differentiability). Let \( I_1 \times \cdots \times I_d = I \subset \mathbb{R}^d \) be closed and bounded, and let \( f \in L^2_\mu(I) \) be a Lipschitz continuous function such that \( \frac{\partial^d f}{\partial x_1^\beta \cdots \partial x_d^\beta} \) exists and is continuous on \( I \) for \( \beta = \sum_{i=1}^{d} \beta_i \). Then the FTT decomposition \((4.7)\) is such that \( \gamma_k(\alpha_{k-1}, \cdot, \alpha_k) \in C^\beta_k(I_k) \) for all \( k \), \( \alpha_{k-1} \) and \( \alpha_k \).

### 4.3. Polynomial approximation of the FTT decomposition

All the theory needed to combine the FTT decomposition with the polynomial approximations described in Section 3 is now in place. We will consider the projection and interpolation approaches separately.

#### 4.3.1. Functional tensor-train projection

Let \( f \in H^k_\mu(I) \) and let \( f_{TT} \) be the rank-\( r \) FTT approximation of \( f \). Applying the projector \((3.4)\) to \( f_{TT} \) yields \( P_N f_{TT} = \sum_{i=0}^{N} \bar{c}_i \Phi_i \), where

\[
\bar{c}_i = \int_I f_{TT}(x)\Phi_i(x) \, d\mu(x) = \sum_{\alpha_0, \ldots, \alpha_d = 1}^{r} \beta_1(\alpha_0, i_1, \alpha_1) \cdots \beta_d(\alpha_d-1, i_d, \alpha_d)
\]

and

\[
\beta_n(\alpha_{n-1}, i_n, \alpha_n) = \int_{I_n} \gamma_n(\alpha_{n-1}, x_n, \alpha_n) \phi_{n}(x_n) \, d\mu_n(x_n).
\]

The spectral expansion of \( f_{TT} \) can thus be obtained by projecting its cores \( \gamma_n(\alpha_{n-1}, x_n, \alpha_n) \) onto univariate basis functions. Furthermore, we immediately have, via \((4.36)\), a tensor-train representation of the expansion coefficients \( C := \{ \bar{c}_i \}_{i=0}^{N} \).

In practice, the projector \( P_N \) is replaced by the discrete projector \( \bar{P}_N \) \((3.7)\), such that the coefficients \( \{ \beta_n \} \) representing projections of the cores are approximated as

\[
\beta_n(\alpha_{n-1}, i_n, \alpha_n) \approx \tilde{\beta}_n(\alpha_{n-1}, i_n, \alpha_n) = \sum_{j=0}^{N_n} \gamma_n(\alpha_{n-1}, x_n^{(j)}, \alpha_n) \phi_{n}(x_n^{(j)}) w_n^{(j)},
\]

where \( \{(x_n^{(j)}, w_n^{(j)})\}_{j=0}^{N_n} \) are appropriate quadrature nodes and weights (e.g., Gauss rules, as described in Section 3) for dimension \( n \). This numerical approximation requires evaluating the cores of the FTT decomposition at the quadrature points. But these values \( \gamma_n(\alpha_{n-1}, x_n^{(j)}, \alpha_n) \) are in fact the cores of the discrete TT representation of \( f(X) \)—i.e., the tensor formed by evaluating \( f \) on the tensor grid of quadrature points. Of course, we never wish to form this tensor explicitly; instead, we obtain its TT approximation via the TT-DMRG-cross algorithm. The end result of this procedure can be viewed as the TT representation \( C_{TT} \) of the spectral coefficient tensor \( C \). The computational procedure is summarized in Algorithm 1.

Once Algorithm \((4.1)\) (FFT-projection-construction) has been run, the spectral TT approximation can be evaluated at an arbitrary point \( y = \{y_1, \ldots, y_d\} \in I \) by the procedure described in Algorithm 2.
and let the matrix be a
\[ (4.39) \]
cost. We will first consider linear interpolation, using the notation of Section 3.2. Let\[ \text{Algorithm 1 FTT-projection-construction} \]
\begin{align*}
\text{Require:} & \quad f : I \rightarrow \mathbb{R}, \text{measure } \mu = \prod_{i=1}^{d} \mu_{n}, \text{integers } N = \{N_{n}\}_{n=1}^{d} \text{ denoting the polynomial degrees of approximation, tolerance } \varepsilon. \\
\text{Ensure:} & \quad \mathcal{C}_{TT}(i_{1}, \ldots, i_{d}) = \sum_{\alpha_{0},\ldots,\alpha_{d}=1}^{r} \beta_{1}(\alpha_{0}, i_{1}, \alpha_{1}) \cdots \beta_{d}(\alpha_{d-1}, i_{d}, \alpha_{d}), \quad \text{the TT-decomposition of the tensor of expansion coefficients.} \\
\text{Construct the set of basis functions } & \quad \left\{ \{\phi_{i_{n}, n}\}_{i_{n}=0}^{N_{n}} \right\}_{n=1}^{d} \text{ with respect to } \mu \\
\text{Determine the quadrature points and weights } & \quad \{(x_{n}, w_{n})\}_{n=1}^{d}, \quad x_{n} = \{x_{n}(i)\}_{i=0}^{N_{n}}, \quad w_{n} = \{w_{n}(i)\}_{i=0}^{N_{n}} \\
\text{Construct the } & \quad \mathbf{A}_{TT} \text{ approximation of } f \left( \times_{j=1}^{d} x_{j} \right) \text{ through } \text{TT-DMRG-cross}, \\
\text{returning the cores } & \quad \left\{ G_{n} \right\}_{n=1}^{d} \text{ and the associated TT-ranks } r. \\
\text{for } & \quad n := 1 \text{ to } d \\
\text{for all } & \quad (\alpha_{n-1}, \alpha_{n}) \in [0, r_{n-1}] \times [0, r_{n}] \text{ do} \\
& \quad \hat{\beta}_{n}(\alpha_{n-1}, i_{n}, \alpha_{n}) = \sum_{j=0}^{N_{n}} G_{n}(\alpha_{n-1}, j, \alpha_{n}) \phi_{i_{n}, n}(x_{n}(j)) w_{n}(j) \\
\text{end for} \\
\text{end for} \\
\text{return } & \quad \left\{ \hat{\beta}_{n}(\alpha_{n-1}, i_{n}, \alpha_{n}) \right\}_{n=1}^{d}
\end{align*}
\[ \text{Algorithm 2 FTT-projection-evaluation} \]
\begin{align*}
\text{Require:} & \quad \text{Cores } \left\{ \hat{\beta}_{n}(\alpha_{n-1}, i_{n}, \alpha_{n}) \right\}_{n=1}^{d} \text{ obtained through } \text{FTT-projection-construction, set of } \quad N_{y} \text{ evaluation points collected in the } N_{y} \times d \text{ matrix } Y = \{y_{1}, \ldots, y_{d}\} \subset I. \\
\text{Ensure:} & \quad \text{The polynomial approximation } \mathbf{P}_{N} f_{TT}(y) \text{ of } f(y) \\
\text{for } & \quad n := 1 \text{ to } d \\
\text{for all } & \quad (\alpha_{n-1}, \alpha_{n}) \in [0, r_{n-1}] \times [0, r_{n}] \text{ do} \\
& \quad \hat{G}_{n}(\alpha_{n-1}, \cdot, \alpha_{n}) = \sum_{j=0}^{N_{n}} \hat{\beta}_{n}(\alpha_{n-1}, j, \alpha_{n}) \phi_{j, n}(y_{n}) \\
\text{end for} \\
\text{end for} \\
\text{return } & \quad \mathbf{P}_{N} f_{TT}(Y) := \left\{ \mathbf{B}_{TT}(i, \ldots, i) \right\}_{i=1}^{N_{y}}
\end{align*}

By Theorems 4.7 and 4.9, the convergence of the spectral expansion depends on the regularity of \( f \). Let \( f \in \mathcal{H}^{k}(I) \) for \( k > d \). Then:
\[
\| f - \mathbf{P}_{N} f_{TT} \|_{L_{x}^{2}(I)} \leq \| f - f_{TT} \|_{L_{x}^{2}(I)} + \| f_{TT} - \mathbf{P}_{N} f_{TT} \|_{L_{x}^{2}(I)} \\
\leq \| f \|_{\mathcal{H}^{k}(I)} \sqrt{(r+1)^{-1} + C(k)N^{-k}|f_{TT}|_{I, \mu, k}}.
\]

This result shows that the convergence is driven by the selection of the rank \( r \) and the polynomial degree \( N \), and that it improves for functions with increasing regularity. Thus we can efficiently compute the expansion coefficients \( \mathcal{C} \) by (4.38) and obtain an approximation \( \mathbf{P}_{N} f_{TT} \) that converges spectrally.

4.3.2. Functional tensor-train interpolation. Function interpolation can easily be extended to tensors, and the tensor-train format can be exploited to save computation and storage costs. We will first consider linear interpolation, using the notation of Section 3.2. Let \( \mathcal{X} = \times_{j=1}^{d} x_{j} \) be a \( N_{x}^{(1)} \times \cdots \times N_{x}^{(d)} \) grid of candidate interpolation nodes where the function \( f \) can be evaluated, and let the matrix \( Y = \{y_{1}, \ldots, y_{d}\} \) of size \( N_{y} \times d \) represent a set of \( N_{y} \) points where one wishes to
### Algorithm 3 FTT-interpolation-evaluation

**Require:** The ε-accurate $\mathcal{A}_{TT}$ approximation of $f(\mathbf{X})$—possibly obtained by TT-DMRG-cross—where $\mathbf{X} = \times_{j=1}^{d} x_j$ is a grid of points constructed from $\{x_n^{(i)}\}_{n=1}^{d}$, and a set of $N_y$ points $\mathbf{Y} = \{y_1, \ldots, y_d\} \subset \mathbf{I}$

**Ensure:** The interpolated approximation $I_{N_f} f_{TT}(\mathbf{Y})$ or $\Pi_{N_f} f_{TT}(\mathbf{Y})$ of $f(\mathbf{Y})$

Construct list $\{L^{(i)}\}_{i=1}^{d}$ of $N_y \times N_x$ (linear or Lagrange) interpolation matrices from $x_i$ to $y_i$ for $n := 1$ to $d$ do

for all $(\alpha_{n-1}, \alpha_n) \in [0, r_{n-1}] \times [0, r_n]$ do

$G_n(\alpha_{n-1}, \alpha_n) = L^{(n)} G_n(\alpha_{n-1}, \alpha_n)$

end for

end for

$\mathcal{B}_{TT}(i_1, \ldots, i_d) = \sum_{\alpha_{d-1}, \alpha_d=1}^{r} \hat{G}_1(\alpha_0, i_1, \alpha_1) \cdots \hat{G}_d(\alpha_{d-1}, i_d, \alpha_d)$

return $I_{N_f} f_{TT}(\mathbf{Y}) := \{ \mathcal{B}_{TT}(i, \ldots, i) \}_{i=1}^{N_y}$

---

evaluate the approximation of $f$. Define $\mathbf{Y} = \times_{j=1}^{d} y_j$. An approximation of $f(\mathbf{Y})$ can be computed using the interpolation operator (3.9) from the grid $\mathbf{X}$ to the grid $\mathbf{Y}$

\[
(4.40) \quad f(\mathbf{Y}) \simeq (I_{N_f} f)(\mathbf{Y}) = E \mathbf{f}(\mathbf{X}), \quad E = E^{(1)} \otimes \cdots \otimes E^{(d)},
\]

where $E^{(k)}$ is a $N_y \times N_x^{(k)}$ matrix defined by $E^{(k)}(i, j) = e_1^{(k)} (y_k^{(i)})$ as in (3.8), and then extracting only the diagonal of the tensor $f(\mathbf{Y})$: $f(\mathbf{Y}) \simeq \{(I_{N_f} f)(\mathbf{Y})_{i_1, \ldots, i_d}\}_{i=1}^{N_y}$. This leads to multi-linear interpolation on hypercubic elements. If we instead use the FTT approximation $f_{TT}$ in (4.40), we obtain

\[
(4.41) \quad (I_{N_f} f_{TT})(\mathbf{Y}) = E f_{TT}(\mathbf{X}) = E \left[ \sum_{\alpha=0, \ldots, \alpha_d=1}^{r} \gamma_1(\alpha_0, x_1, \alpha_1) \cdots \gamma_d(\alpha_{d-1}, x_d, \alpha_d) \right]
\]

\[
= \sum_{\alpha=0, \ldots, \alpha_d=1}^{r} \beta_1(\alpha_0, y_1, \alpha_1) \cdots \beta_d(\alpha_{d-1}, y_d, \alpha_d),
\]

with

\[
\beta_n(\alpha_{n-1}, y_n, \alpha_n) = E^{(n)} \gamma_n(\alpha_{n-1}, x_n, \alpha_n),
\]

where instead of working with the tensor $E$, we can work with the more manageable matrices $\{E^{(i)}\}_{i=1}^{d}$ (see Algorithm 3). The construction of the approximation in this case corresponds exactly to the application of the TT-DMRG-cross algorithm to $f(\mathbf{X})$ to obtain $\mathcal{A}_{TT}$. The listing of FTT-interpolation-construction is thus omitted. The basis functions (3.8) yield quadratic convergence of the interpolant to the target function. Thus, for $k > d - 1$ and $f \in \mathcal{H}_b^k(1)$,

\[
(4.42) \quad \|f - I_{N_f} f_{TT}\|_{L_b^2(1)} \leq \|f\|_{\mathcal{H}_b^k(1)} \sqrt{\frac{(r+1)^{-1} - (k-1) \nu^{d-1} - 1}{k-1}} + C N^{-2} \|f_{TT}\|_{1, \mu, 2}.
\]

Additionally these basis functions have local support (as opposed to the global support of the polynomials used for the projection) and this prevents the propagation over all the space of errors due to singularities of $f$.

The same approach can be taken for the interpolation with Lagrange polynomials. The interpolating values can be obtained extracting the diagonal $f(\mathbf{Y}) \simeq \{(\Pi_{N_f} f_{TT})(\mathbf{Y})_{i_1, \ldots, i_d}\}_{i=1}^{N_y}$ of

\[
(4.43) \quad f(\mathbf{Y}) \simeq (\Pi_{N_f} f)(\mathbf{Y}) = L f(\mathbf{X}), \quad L = L^{(1)} \otimes \cdots \otimes L^{(d)},
\]
where $L^{(k)}$ is the $N_y \times N_z^{(k)}$ Lagrange interpolation matrix [35]. This interpolation is not carried out directly in high dimension, but we only need to perform one-dimensional interpolations of the cores (see Alg. 3):

\begin{equation}
(\Pi_N f_{TT})(\mathbf{Y}) = L f_{TT}(\mathbf{X}) = \sum_{\alpha_0,\ldots,\alpha_d=1}^r \beta_1(\alpha_0,\mathbf{y}_1,\alpha_1) \cdots \beta_d(\alpha_{d-1},\mathbf{y}_d,\alpha_d),
\end{equation}

\begin{equation}
\beta_n(\alpha_{n-1},\mathbf{y}_n,\alpha_n) = L^{(n)} \gamma_n(\alpha_{n-1},\mathbf{x}_n,\alpha_n).
\end{equation}

The convergence is again dictated by the regularity of the function $f$. For $k > d - 1$ and $f \in \mathcal{H}^k_\mu(I)$:

\begin{equation}
\|f - \Pi_N f_{TT}\|_{L^2(\mathbb{I})} \leq \|f\|_{\mathcal{H}^k_\mu(\mathbb{I})} \sqrt{\frac{(r+1)^{(k-1)}}{k-1} \frac{r^{d-1}-1}{r-1} + C(k)N^{-k}\|f_{TT}\|_{\mathcal{H}^k_\mu(\mathbb{I})}}.
\end{equation}

4.3.3. Spectral TT algorithms. Suppose we have a function $f : \mathbb{I} \to \mathbb{R}$ where $\mathbb{I} = \times_{i=1}^d I_i$ and $I_i \subseteq \mathbb{R}$, for $i = 1, \ldots, d$. We would like to construct an approximation of $f$ and to evaluate this approximation on an independent set of points $\mathbf{Y}$. The algorithm for constructing and evaluating the spectral tensor-train approximation of $f$ proceeds as follows:

1. select a suitable set of candidate nodes $\mathbf{X} = \times_{n=1}^d \mathbf{x}_n$ according to the type of approximation to be constructed
2. construct the approximation using Algorithm [1] for the projection approach or directly using TT-DMRG-cross [48] on $f(\mathbf{X})$ for the interpolation approach
3. evaluate the spectral tensor-train approximation on $\mathbf{Y}$ by Algorithm [2] for the projection approach or by Algorithm [3] for the interpolation approach.

In the following we will refer to the FTT-projection and the FTT-interpolation algorithms as the combination of the two corresponding steps of construction and evaluation.

The practical implementation uses data structures to cache computed values and to store partially computed decompositions. It also fully supports the usage of the Message Passing Interface (MPI) protocol for the parallel evaluation of $f$ during the execution of TT-DMRG-cross.

5. Numerical examples. The Spectral tensor-train decomposition is now applied to several high dimensional functions, with the aim of obtaining a surrogate model of it. The quality of such surrogate models will be evaluated using the relative $L^2$ error:

\begin{equation}
\|f - L f_{TT}\|_{L^2(I)} = \sqrt{\frac{\int_I (f - L f_{TT})^2 d\mu}{\int_I f^2 d\mu}},
\end{equation}

where $L$ is one of the projection ($P_N$) or interpolation ($I_N$, $\Pi_N$) operators. This high dimensional integral is estimated using the Monte Carlo estimator, with the number of samples driven by the target relative tolerance of $10^{-2}$.

\footnote{In Sec. 5.1.2 $\|f - L f_{TT}\|_{L^2(I)}$ was used in place of (5.1) for consistency with the results obtained in [10].}
5.1. Genz functions. The Genz functions \cite{17,18} are a set of functions, defined on $[0, 1]^d$, frequently used to estimate the properties of approximation schemes. They are defined as follows:

\begin{align}
\text{oscillatory: } f_1(x) &= \cos\left(2\pi w_1 + \sum_{i=1}^d c_i x_i\right) \\
\text{product peak: } f_2(x) &= \prod_{i=1}^d \left(c_i^{-2} + (x_i + w_i)^2\right)^{-1} \\
\text{corner peak: } f_3(x) &= \left(1 + \sum_{i=1}^d c_i x_i\right)^{-(d+1)} \\
\text{Gaussian: } f_4(x) &= \exp\left(-\sum_{i=1}^d c_i^2(x_i - w_i)^2\right) \\
\text{continuous: } f_5(x) &= \exp\left(-\sum_{i=1}^d c_i^2|x_i - w_i|\right) \\
\text{discontinuous: } f_6(x) &= \begin{cases} 0 & \text{if any } x_i > w_i \\ \exp\left(\sum_{i=1}^d c_i x_i\right) & \text{otherwise} \end{cases}
\end{align}

Except for the “discontinuous” function, the parameters $w$ are drawn uniformly from $[0, 1]$. These parameters act as a shift for the function. For the “discontinuous” function $w$ determines the position of the hyperplane defining a discontinuity of the function. If $w$ was drawn uniformly also in this case, the probability of being in the non-zero region of the function would decrease exponentially with the dimension. This would make it very hard to obtain an error estimate for our approximation with Monte Carlo method. Then we impose that for $x \sim U([0, 1]^d)$, $P[x > w] = 1/2$. This was achieved selecting $w \sim \text{Beta}(\alpha, \beta)$, where $\beta = 1$ and $\alpha = \exp\left(\frac{\log(1/2)}{d}\right) / \left(1 - \exp\left(\frac{\log(1/2)}{d}\right)\right)$.

The parameters $c$ are drawn uniformly from $[0, 1]$ and then normalized to $d\parallel c\parallel_1 = b_j$, for $j$ indexing the six Genz functions. The “difficulty” of the function increases monotonically with $b_j$ and $e_j$ is a scaling constant used for different dimensions. The parameters $e_j$ are defined as suggested in \cite{17,18}, while $b_j$ are selected in order to obtain the same test functions used for $d = 10$ in \cite{1}. These are listed in table 5.1.

To compare the results obtained by functional tensor-train projection with the Smolyak pseudospectral sparse grid approximation \cite{10}, we also consider the normalization $\parallel c\parallel_1 = a_j$ for $d = 5$ with values listed in table 5.1.

The experiments will be performed picking 30 different sets of parameters $w$ and $c$ for each Genz function and looking at the $L^2$ error (5.1) with respect to the number of function evaluations needed to construct an approximation based on the functional tensor-train projection or interpolation, with a desired order or a desired refinement respectively. Both the error estimate and the number of function evaluations can vary depending on the particular function at hand. In particular the number of function evaluations is driven by the procedure for obtaining a pointwise tensor-train approximation on the tensor grid using the TT-DMRG-cross algorithm (see sec. 2.3).
Fig. 5.1: Functional tensor-train projection approximation of the Genz functions. For exponentially increasing polynomial order ($2^i - 1$ for $i = 1, \ldots, 4$) and for different dimensions, 30 Genz functions have been constructed and approximated using the FTT-projection algorithm. The scattered dots show the $L^2$ error and the number of function evaluations needed for each of these realizations. The circled dots represent the mean $L^2$ error and mean number of function evaluations for increasing polynomial order.
Fig. 5.2: FTT-projection approximation of the Genz functions. For exponentially increasing polynomial order \( 2^i - 1 \) for \( i = 1, \ldots, 4 \) and for different dimensions, 30 Genz functions have been constructed and approximated using the FTT-projection algorithm. The dots show the number of function evaluations with respect to the polynomial order selected.
The target accuracy of the TT-DMRG-cross approximation in terms of Frobenius norm is set to $\varepsilon_{\text{rnd}} = 10^{-10}$ to be conservative.

5.1.1. Functional tensor-train projection on the Genz functions. In the next tests different dimensions will be considered ranging between 10 and 100. The “corner peak” function was tested up to $d = 15$ due to the higher computational effort required to build the approximation. The decay of the singular values of this function is very slow, leading to an increased sampling. For the “product peak” function we could not run the tests for $d > 20$ because $f_2 \to 0$ as $d$ increases, leading to a loss of machine precision.

Figure 5.1 shows the convergence rate of the FTT-projection approximation on the six Genz functions for exponentially increasing polynomial order ($2^i - 1$ for $i = 1, \ldots, 4$). The quadrature points used are Gauss points. Due to the interchangeability of the dimensions in the Genz functions, the single realizations are more scattered for the low-dimensional functions, being these defined by a smaller number of random parameters.

As expected we obtain the spectral convergence rate on the smooth functions 1-4. On the “continuous” Genz function the convergence is only quadratic, due to the first order discontinuity in its definition. The approximation to the “discontinuous” function requires more function evaluations compared to the other functions: the reason lays in the fact that all the other functions have an exact low rank representation, meaning that the singular values rapidly become zero, leading to no information loss when the truncation is performed in order to select the TT-ranks. The “corner peak” function, instead, couples all the variables with the outer exponentiation, leading to a slower decay of the singular values $\sigma(\alpha)$ and to the necessity of increasing the TT-ranks in order to meet the accuracy requirements. The relation between number of function evaluations and the order of the polynomial basis used is shown in figure 5.2. Again, the effect of not being of low-rank, penalizes the performances on the “corner peak” function.

5.1.2. Comparison with sparse grid pseudospectral approximation. The goal of this numerical example is to compare our results to the fully adaptive Smolyak sparse grid pseudospectral approximation [10], where the number of function evaluations is increased by increasing the available computational time. For this test we will consider only the first four smooth Genz functions with $d = 5$ as done in [10]. Figure 5.3 shows that the functional tensor-train projection outperforms the pseudospectral approximation in all the tests where an exact low-rank decomposition of the function exists. The “corner peak” function doesn’t have an exact low-rank decomposition and spectral-tensor train is outperformed by the pseudospectral approximation in this case. It is fair to notice however that the fully adaptive pseudospectral approximation performs an anisotropic order adaptation with respect to the dimensions, i.e. it can use different orders on different dimensions. This is a feature that is not yet available in our implementation of spectral tensor-train, thus the increase of order is isotropic, leading to an excessive refinement in certain directions, leaving room for future improvements.

5.2. Modified Genz functions. It is noticeable, from figure 5.1 that the approximations tend to get easier as the dimension is increased. This is due to the fact that the Genz functions were not designed to be used for very high dimensions. As an example, consider the “Gaussian” function $f_4$. It has the rank one representation:

$$f_4(x) = \exp \left( - \sum_{i=1}^{d} c_i^2 (x_i - w_i)^2 \right) = \prod_{i=1}^{d} \exp \left( -c_i^2 (x_i - w_i)^2 \right).$$

The $c$ vector is normalized so that $\|c\|_1 = b_j/d_j$. Then, for $d \to \infty$ and for the values of $e_j$ and $b_j$ listed in table 5.1, $c_i \to 0$ and $f_4 \to 1$. This means that the higher is the dimension, the closer the function is to be a constant and thus easier to be approximated.
Fig. 5.3: Functional tensor-train projection approximation and Smolyak sparse grid pseudospectral approximation of the Genz functions. For increasing accuracy 30 Genz functions have been constructed and approximated by the two methods. The scattered dots show the $L^2$ error and the number of function evaluations needed for each of these realizations. The circled/square dots represent the mean $L^2$ error and mean number of function evaluations for increasing accuracy levels.
Fig. 5.4: Functional tensor-train projection approximation of the modified Genz functions. For exponentially increasing polynomial order ( $2^i - 1$ for $i = 1, \ldots, 4$ ) and for different dimensions, 30 modified Genz functions have been constructed and approximated using the FTT-projection algorithm. The scattered dots show the $L_2$ error and the number of function evaluations needed for each of these realizations. The circled dots represent the mean $L_2$ error and mean number of function evaluations for increasing polynomial order.
5.2.1. Functional tensor-train projection on the modified Genz functions. As a mean of comparison with the original Genz functions, we consider the performances of the functional tensor-train projection on their modified version. Figure 5.4 shows the convergence rate of the surrogate function with respect to the number of function evaluation, for increasing polynomial order. A comparison with figure 5.1 shows that the tests on the Modified Genz functions are more informative about the method with respect to the original functions, because they don’t become easier with the increase of dimensions. Again the spectral convergence is obtained on the smooth functions. The higher scattering of the points in the approximation of the “corner peak” function is due to the absence of an analytic low-rank representation for such function, and thus the introduction of truncation in the tensor-train decomposition.

5.2.2. Functional tensor-train interpolation on the modified Genz functions. The linear FTT-interpolation has been tested on all the modified Genz functions, with an exponentially increasing number – from $2^1$ to $2^7$ – of uniformly distributed interpolating points and for different dimensions, 30 Genz functions have been constructed and approximated by the tensor-train linear interpolation. The scattered dots show the $L^2$ error and the number of function evaluations needed for each of these realizations. The circled dots represent the mean $L^2$ error and mean number of function evaluations for increasing grid refinements.

Fig. 5.5: Functional tensor-train linear interpolation of the “continuous” and “discontinuous” modified Genz functions. For exponentially increasing number – from $2^1$ to $2^7$ – of uniformly distributed interpolating points and for different dimensions, 30 Genz functions have been constructed and approximated by the tensor-train linear interpolation. The scattered dots show the $L^2$ error and the number of function evaluations needed for each of these realizations. The circled dots represent the mean $L^2$ error and mean number of function evaluations for increasing grid refinements.

We would instead like to test the performance of the spectral tensor-train approximation on a more realistic set of example functions, whose “difficulty” grows with the dimension. To this end, we use the definition (5.2) of the Genz functions, but we refrain from normalizing the coefficients $c \sim U([0,1])$. This leads to functions that don’t degenerate to constants in high dimensions, and thus can be used for testing purposes at higher dimensions than the original Genz functions.

The Lagrange FTT-interpolation has also been tested for all the modified Genz functions. We omit the results here because they are in line with the results obtained with the FTT-projection showed in Fig. 5.1.
5.3. **FTT-projection and mixed Fourier modes.** It is now understood that the approximation of multidimensional functions with sparse grids is exact when the function’s Fourier coefficients are non-zero only for the set of admissible multi-indices included in the sparse grid construction [10, 11]. The convergence of the approximation deteriorates when the decay of the Fourier coefficients is slow for mixed modes.

We construct two ad-hoc functions to highlight some properties of the FTT-projection, when approximating functions with different types of decay in their Fourier coefficients. Let us consider functions defined on $I = I_1 \times \cdots \times I_d$ where $I_i = [-1, 1]$. On this hypercube we consider the sub-cube $I_{j_1} \times \cdots \times I_{j_c}$, where $J = \{j_i\}_{i=1}^c \subseteq \{1, \ldots, d\}$. For every index in $J$, we select $\{n_{j_1}, \ldots, n_{j_c}\} > 0$ to be the maximum order of polynomials included in the functions along the $i$-th direction. The functions will then be defined as follows:

\[
\begin{align*}
    f_1(x) &= \prod_{k=1}^{c} \phi_{j_k}(x_{j_k}) , \\
    f_2(x) &= \sum_{i_{j_1}=0}^{n_{j_1}} \cdots \sum_{i_{j_c}=0}^{n_{j_c}} \exp \left(-i^T \Sigma i\right) \prod_{k=1}^{c} \phi_{j_{ik}}(x_{j_k}) ,
\end{align*}
\]

where $\Sigma$ is a $c \times c$ matrix defining the level of interaction between different dimensions, $\{\phi_{j_{ik}}\}_{i_{j_k}=1}^{n_{j_k}}$ are chosen to be the normalized Legendre polynomials, $i = (i_{j_1}, \ldots, i_{j_c})^T$ and the $\phi_{j_k}$ are possibly high order polynomials. To simplify the notation, we will set $n_{j_k} = n$ for all $j_k$.

The function $f_1$ is a function with one single high mixed Fourier mode as shown in figure 5.6a. In spite of the high polynomial order, the rank of the function is correctly estimated to be 1 and thus very few sampling points are needed in order to recover the required precision. This highlights that, on the contrary of sparse grids, the spectral tensor-train does not discard basis functions in its construction, but it uses always a fully tensorized set of basis functions.
SPECTRAL TENSOR-TRAIN DECOMPOSITION

\[ f(x, y) = \exp \left( -\frac{|x - x_0|^2}{2l^2} \right) \]

Let \( d = 2 \), \( x_0 = [0.2, 0.2] \) and \( l = 0.05 \). The function shows an off-centered peak as shown in figure 5.7a. The points used by TT-DMRG-cross (with accuracy \( \varepsilon = 10^{-10} \)) are shown on the same figure,

\[ \Sigma = \begin{bmatrix} 1 & -0.9 \\ -0.9 & 1 \end{bmatrix} \]

and \( J = [0, 1] \). The decay of the coefficients, as estimated using the FTT-projection, is shown in figure 5.6b. The function has an high TT-rank and this leads to the complete sampling of the space. We can use this function also to experiment on what is called the ordering problem of the TT-decomposition. We let \( d = 5 \) and use different combinations of indices in \( J \). If \( J \) contains two neighboring dimensions, \( J = [1, 2] \) in the example, the TT-ranks of the decomposition will be \( r = [1, 1, 11, 1, 1] \), where the maximum is attained between the cores \( G_1 \) and \( G_2 \). If we consider \( J \) containing non-neighboring dimensions, \( J = [0, 4] \) in the example, we practically obtain the same function, with reordered dimensions. In this case the TT-ranks will be \( r = [1, 11, 11, 11, 11, 1] \). This happens due to the hierarchical construction of the TT-decomposition, where information can be propagated only from one core to the next one. The example shows that the only consequence of a wrong ordering choice is that it can lead to an increased number of function evaluations, which grows with \( r^2 \). This however does not affect the accuracy of the approximation.

5.4. Resolution of local features. It is often the case that the modeled function presents important local features which need to be resolved accurately. An a priori clustering of nodes is not possible because the location of such feature is unknown. The TT-DMRG-cross algorithm overcomes this problem, because it adaptively selects the nodes that are relevant for the approximation, thus exploring the space with an increasing knowledge about the features of the function. As an explanatory example, consider

\[ f(x) = \exp \left( -\frac{|x - x_0|^2}{2l^2} \right) \]

Fig. 5.7: The left figure shows the off-centered local feature of (5.5). The white and black dots show the candidate points where the function has been evaluated. The black dots show the points that are used in the final TT-DMRG-cross approximation. TT-DMRG-cross detects the feature and clusters the nodes around it, in order to obtain maximum accuracy (\( \varepsilon = 10^{-10} \)). The right figure shows the same test for \( d = 3 \).
where the white dots are the points used on the way to the final approximation, while the black dots are the points retained in the final approximation. Figure 5.7 shows the set of points used for $d = 3$ and $x_0 = [0.2, 0.2, 0.2]$. The same kind of clustering is observed.

### 5.5. Elliptic equation with random input data

Here we consider the classical Poisson’s equation defined on the unit square $\Gamma = [0, 1] \times [0, 1]$

\[
\begin{aligned}
\begin{cases}
-\nabla \cdot (\kappa(x, \omega) \nabla u(x, \omega)) = f(x, \omega) & \text{in } \Gamma \times \Omega \\
u(x, \omega) = 0 & \text{on } \partial \Gamma \times \Omega,
\end{cases}
\end{aligned}
\]

(5.6)

where $f(x, \omega) = 1$ is a deterministic load, and $\kappa$ is a log-normal random field defined on the probability space $(\Omega, \Sigma, \mu)$ by

\[
\kappa(x, \omega) = \exp \left( \frac{g(x, \omega)}{10} \right), \quad g(x, \omega) \sim \mathcal{N}(0, C_g(x, y)).
\]

(5.7)

We characterize the normal random field $g \in L_2^2(\Omega; L^\infty(\Gamma))$ by the squared exponential covariance:

\[
C_g(x, y) = \int_{\Omega} g(x, \omega) g(y, \omega) d\mu(\omega) = \exp \left( -\frac{||x - y||^2}{2l^2} \right),
\]

(5.8)

where $l > 0$ determines the spatial correlation length of the field. We decompose the random field through the Karhunen-Loève (KL) expansion.

\[
C_g(x, y) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} \chi_i(x) Y_i(\omega),
\]

(5.9)

where $Y_i \sim \mathcal{N}(0, 1)$ and $\{\lambda_i, \chi_i(x)\}_{i=1}^{\infty}$ are the eigenvalues and eigenfunctions of the eigenvalue problem $\int_{\Omega} C_g(x, y) \chi_i(y) dy = \lambda_i \chi_i(x)$. The KL-expansion is truncated in order to retain the 95% of the total variance ($\text{Var}[g(x, \omega)] = 1$), i.e. we find $d \in \mathbb{N}^+$ such that $\sum_{i=1}^{d} \lambda_i \geq 0.95$. With a correlation length of $l = 0.25$ we use $d = 12$ terms in the KL-expansion. Figure 5.8a shows one realisation of the random field (5.7), computed using the selected parameters for the KL-expansion.

The use of the KL-expansion allows (5.6) to be turned into a parametric problem, where we seek the solution $u \in L^2(\Omega) \times L^2(\partial \Omega; \mathbb{R}^d)$. Here we will focus on the construction of a surrogate of $u(x_0, Y)$, for $x_0 = (0.75, 0.25)$.

The surrogate is constructed using the FTT-projection with Hermite polynomials as basis functions. Figure 5.8b shows the convergence, in terms of the $L^2$ error (5.1), for orders 0, 1, 3 and 7 for different target accuracies. These accuracies are driven by the tolerances that are set in the TT-DMRG-cross algorithm, and they represent the accuracy with which the discrete tensor of values is approximated by the TT-decomposition. We can see that the accuracy of the surrogate improves spectrally until the target accuracy is reached. After this happens, an increase in the order of the surrogate doesn’t provide any more improvement and the convergence plot flattens at the target accuracy level.

### 6. Conclusions

This paper presents a novel and rigorous construction of the Spectral tensor-train decomposition, that can be used for the approximation of high-dimensional functions. The method aims at tackling the curse of dimensionality for functions with sufficient regularity, exploiting the low-rank representation of the approximated function, and at attaining spectral convergence, by the use of polynomial approximation.

We present an iterative procedure to decompose an arbitrary function $f \in L^2_\mu(\mathcal{I})$, obtaining a format that we call the functional tensor-train decomposition, to distinguish it from the already studied discrete tensor-train decomposition. The construction of the surrogate is based on the existence of the singular value decomposition of Hilbert-Schmidt kernels in $L^2_\mu(\mathcal{I})$ and on the regularity...
properties of the function (c.f. Thm. 4.7). This regularity will be carried on by the singular functions of the decomposition (c.f. Thm. 4.9 and 4.11), leading to the same convergence rate that would be obtained if we applied the polynomial approximation to the high-dimensional $f$.

The tensor-train decomposition \cite{45} obtained through the TT-DMRG-cross algorithm \cite{48} leads to a memory and computational complexity that scale linearly with the dimensionality of the function. The theory of polynomial approximation is added on top of the discrete representation obtained by TT-DMRG-cross, and provides an accurate approximation that converges spectrally on smooth functions. The user is required to select the polynomial order of the approximation and the overall accuracy required. The latter tolerance will drive the amount of dimensional interaction described by the approximation and ultimately the number of function evaluations, which will grow mildly for functions with a fast decay of their singular values.

Unlike in sparse grid pseudospectral approximation, the method doesn’t make any a priori assumption in the choice of the basis for the separation of the space $L^2_{\mu}(I)$. Instead, it uses the singular functions of $f$, which are optimal. The choice of a polynomial basis is made during the projection of the singular functions $\gamma_k(\alpha_k-1, \cdot, \alpha_k) \in L^2_{\mu}(I_k)$ onto the space spanned by such fully tensorized polynomials. This approach also permits to resolve local features not positioned at the center of the domain, by clustering the evaluation points close to the feature.

In some cases, the performances of the method are dependent on the ordering of the dimensions. This results only in a higher number of function evaluations, although still linear in $d$, but does not compromises the quality of the approximation. Research in the direction of finding an optimal ordering a priori is a topic of ongoing work.

The results from this work pave the way to an adaptive spectral tensor-train decomposition: the smoothness properties of the singular functions can in fact be used as an indicator for the necessity of increasing the polynomial order on each dimension. This will allow the complete automation of the construction of spectral tensor-train surrogates.

The results in this work have been obtained using the open-source Python library for spectral
tensor-train decomposition that is made available online— including examples from this paper.

Acknowledgments. The authors would like to thank Jan Hesthaven, Alessio Spantini, Florian Augustin, and Patrick Conrad for fruitful discussions on this topic and for providing many useful comments on the paper. We would also like to thank Dmitry Savostyanov for calling our attention to the TT-DMRG-cross algorithm, just after reading a preprint of this work.

Appendix A. Hölder continuity and the Smithies condition. In Section 4.2 we use a result by Smithies [53, Thm. 14] to prove the boundedness of the weak derivatives of the cores of the FTT-decomposition. The condition under which Smithies’ result hold is:

Definition A.1 (Smithies’ integrated Hölder continuity). Let \( K(s,t) \) defined for \( s,t \in [a,b] \).
Without loss of generality, let \( a = 0 \) and \( b = \pi \). For \( r > 0 \), let

\[
K^{(i)}(s,t) = \frac{\partial^i K(s,t)}{\partial s^i}, \quad 0 < i \leq r
\]

and let \( K^{(1)}, \ldots, K^{(r−1)} \) exist and continuous. Let \( K^{(r)} \in L^p(s) \) a.e. in \( t \) and \( 1 < p \leq 2 \). The integrated Hölder continuity, with either \( r > 0 \) and \( \alpha > 0 \) or \( r = 0 \) and \( \alpha > \frac{1}{p} − \frac{1}{2} \), holds for \( K \) if there exists \( A > 0 \) such that:

\[
(A.2) \quad \int_0^\pi \left\{ \int_0^\pi \left| K^{(r)}(s+\theta,t) − K^{(r)}(s-\theta,t) \right|^p \, ds \right\}^{\frac{2}{p}} \, dt \leq A|\theta|^{2\alpha}.
\]

This definition is of difficult interpretation. Furthermore, in the scope of this work, we are interested in the case \( r = 0 \). A simpler, but not equivalent, definition is the one mentioned in [56]:

Definition A.2 (Hölder continuity almost everywhere). Let \( K(s,t) \) defined for \( s,t \in [a,b] \). \( K \) is Hölder continuous a.e. with exponent \( \alpha > 0 \) if there exists \( C > 0 \) such that

\[
(A.3) \quad |K(s+\theta,t) − K(s-\theta,t)| \leq C|\theta|^\alpha
\]

almost everywhere in \( t \).

For the sake of simplicity, we show that:

Proposition A.3. The Hölder continuity a.e. is a sufficient condition for the Smithies’ integrated Hölder continuity.

Proof. Let \( K \in L^p(s) \) for almost all \( t \), \( 1 < p \leq 2 \). For \( \alpha > \frac{1}{p} \), let \( K \) be Hölder continuous a.e. in \( t \). Then:

\[
(A.4) \quad \int_0^\pi \left\{ \int_0^\pi \left| K^{(r)}(s+\theta,t) − K^{(r)}(s-\theta,t) \right|^p \, ds \right\}^{\frac{2}{p}} \, dt \leq \int_0^\pi \left\{ \int_0^\pi C^p |\theta|^\alpha \, ds \right\}^{\frac{2}{p}} \, dt
\]

\[
= C^2 \pi^{\frac{2}{p}} |\theta|^{2\alpha} \leq C^2 \pi^{\frac{3}{p}} |\theta|^{2\alpha} = A|\theta|^{2\alpha}
\]

where we recognize the bound \((A.2)\) of the Smithies’ integrated Hölder continuity.

Appendix B. Proofs of auxiliary results for Theorem 4.7.

Proof. [Proof of lemma 4.5] By definition of Sobolev norm, seminorm and weak derivative \( D^i \):

\[
|J|_{I_1 \times I_2, \mu, k}^2 \leq \|J\|_{K_E(t_1 \times I_1)}^2 = \sum_{|i|=0}^k \|D^i(f(x,y), f(\bar{x}, y))\|_{L_2(t_1 \times I_1)}^2
\]

(B.1)

\[
= \sum_{|i|=0}^k \|D^{i_1,0}f(x,y), D^{i_2,0}f(\bar{x},y)\|_{L_2(t_1 \times I_1)}^2.
\]

https://pypi.python.org/pypi/TensorToolbox/
where \( i \) is a two dimensional multi-index. Using the Cauchy-Schwarz inequality, it holds:

\[
\|D^{i_0} f(x, y)\|_{L^2(I_1 \times I_1)}^2 \leq \|D^{i_0} f(x, y)\|_{L^2(\bar{I})}^2 \|D^{i_0} f(x, y)\|_{L^2(\bar{I})}^2
\]

Let now \( j \) and \( l \) be two \( d \)-dimensional multi-indices, then \([B.1]\) can be bounded by

\[
|J|^2_{I_1 \times I_1, \mu, k} \leq \|J\|^2_{\mathcal{H}^k(I_1 \times I_1)} \leq \sum_{|i|=0}^k \|D^i f(x, y)\|_{L^2(I_1 \times I_1)}^2 \|D^i f(x, y)\|_{L^2(I_1 \times I_1)}^2 \
\]

\[
\leq \sum_{|i|=0}^k \sum_{|l|=0}^k \|D^j f(x, y)\|_{L^2(I_1 \times I_1)}^2 \|D^j f(x, y)\|_{L^2(I_1 \times I_1)}^2 \leq \|f\|^2_{\mathcal{H}^k(I_1 \times I_1)}.
\]

Since \( \|J\|_{\mathcal{H}^k(I_1 \times I_1)} \leq \|f\|^2_{\mathcal{H}^k(I_1 \times I_1)} < \infty \) by assumption, then \( J \in \mathcal{H}^k(I_1 \times I_1) \).

**Proof.** [Proof of lemma 4.0] Using the definition of Sobolev norm and theorem 4.11,

\[
\|\varphi_{i+1}(\alpha_{i+1})\|_{\mathcal{H}^k(I)}^2 = \sum_{|j|=0}^k \|D^j \varphi_{i+1}(\alpha_{i+1})\|_{L^2(I)}^2 \\
\leq \sum_{|j|=0}^k \frac{1}{\lambda_{i+1}(\alpha_{i+1})} \|\gamma_{i+1}(\alpha_{i+1})\|_{L^2(I_1 \times I_1)}^2 \|D^j \sqrt{\lambda_i} \varphi_i(\alpha_i)\|_{L^2(\bar{I})}^2 \\
= \frac{1}{\lambda_{i+1}(\alpha_{i+1})} \sum_{|j|=0}^k \|D^j \sqrt{\lambda_i} \varphi_i(\alpha_i)\|_{L^2(I)}^2 \\
= \frac{1}{\lambda_{i+1}(\alpha_{i+1})} \left\| \frac{1}{\sqrt{\lambda_i}} \varphi_i(\alpha_i) \right\|_{\mathcal{H}^k(I)}^2.
\]

This argument can be used recursively,

\[
\|\varphi_{i+1}(\alpha_{i+1})\|_{\mathcal{H}^k(I)}^2 \leq \frac{1}{\lambda_{i+1}(\alpha_{i+1})} \left\| \sqrt{\lambda_i} \varphi_i(\alpha_i) \right\|_{\mathcal{H}^k(I)}^2 \\
= \frac{1}{\lambda_{i+1}(\alpha_{i+1})} \lambda_i(\alpha_i) \|\varphi_i(\alpha_i)\|_{\mathcal{H}^k(I)}^2 \\
\leq \cdots \leq \frac{1}{\lambda_{i+1}(\alpha_{i+1})} \|f\|^2_{\mathcal{H}^k(I)},
\]

completing the proof.  

**REFERENCES**

[1] Volkmar Barthelmann, Erich Novak, and Klaus Ritter, High dimensional polynomial interpolation on sparse grids, Advances in Computational Mathematics, 12 (2000), pp. 273–288.

[2] C Bernardi and Y Maday, Polynomial interpolation results in Sobolev spaces, Journal of computational and applied mathematics, (1992).

[3] J P Boyd, Chebyshev and Fourier Spectral Methods: Second Revised Edition, Dover Books on Mathematics, Dover Publications, New York, 2 ed., 2001.

[4] Susanne C. Brenner and L. Ridgway Scott, The Mathematical Theory of Finite Element Methods, vol. 15 of Texts in Applied Mathematics, Springer New York, New York, NY, 2008.

[5] R. Bro, Multi-way analysis in the food industry: models, algorithms, and applications, PhD thesis, Universiteit van Amsterdam, 1998.
[6] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang, Spectral Methods - Fundamentals in Single Domains, Scientific Computation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[7] JD Carroll and JJ Chang, Analysis of individual differences in multidimensional scaling via an N-way generalization of “Eckart-Young” decomposition, Psychometrika, 35 (1970).

[8] Ali Civril and Malik Magdon-Ismail, On selecting a maximum volume sub-matrix of a matrix and related problems, Theoretical Computer Science, 410 (2009), pp. 4801–4811.

[9] Francisco Chinesta, Roland Keunings, and Adrien Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations, SpringerBriefs in Applied Sciences and Technology, Springer International Publishing, Cham, 2014.

[10] Patrick R. Conrad and Youssef M. Marzouk, Adaptive Smolyak Pseudospectral Approximations, SIAM Journal on Scientific Computing, 35 (2013), pp. A2643–A2670.

[11] Paul G. Constantine, Michael S. Eldred, and Eric T. Phipps, Sparse pseudospectral approximation method, Computer Methods in Applied Mechanics and Engineering, 229-232 (2012), pp. 1–12.

[12] Sergey Dolgov, Boris N. Khoromskij, Alexander Litvinenko, and Hermann G. Matthies, Computation of the Response Surface in the Tensor Train data format, arXiv preprint arXiv:1406.2816, (2014), p. 28.

[13] Ali Reza Doostan and Gianluca Iaccarino, A least-squares approximation of partial differential equations with high-dimensional random inputs, Journal of Computational Physics, 228 (2009), pp. 4332–4345.

[14] Ali Reza Doostan, AbdoulAhad Validi, and Gianluca Iaccarino, Non-intrusive low-rank separated approximation of high-dimensional stochastic models, Computer Methods in Applied Mechanics and Engineering, 263 (2013), pp. 42–55.

[15] Mike Espig, Wolfgang Hackbusch, Alexander Litvinenko, Hermann G. Matthies, and Philipp Wähnert, Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats, Computers & Mathematics with Applications, 67 (2014), pp. 818–829.

[16] Walter Gautschi, Orthogonal Polynomials: Computation and Approximation, Numerical Mathematics and Scientific Computation, Oxford University Press, 2004.

[17] A Genz, Testing multidimensional integration routines, Proc. of international conference on Tools, methods and languages for scientific and engineering computation, (1984).

[18] A package for testing multiple integration subroutines, Numerical Integration, (1987).

[19] Loïc Giraldi, Alexander Litvinenko, Dishi Liu, Hermann G. Matthies, and Anthony Nouy, To be or not to be intrusive? The solution of parametric and stochastic equations - the 'plain vanilla' Galerkin case, arXiv: 1309.1617, (2013), pp. 1–24.

[20] GH Golub and JH Welsch, Calculation of Gauss quadrature rules, Mathematics of Computation, (1969), pp. 221–230.

[21] S Goreinov and I Oseledets, How to find a good submatrix, in Matrix methods: Theory, Algorithms and Applications, World Scientific Publishing Co. Pte. Ltd., Singapore, 2010, pp. 247–256.

[22] SA Goreinov, NL Zamarashkin, and EE Tyrtyshnikov, Pseudo-skeleton approximations by matrices of maximal volume, Mathematical Notes, 62 (1997), pp. 619–623.

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

[24] Lars Grasedyck, Daniel Kressner, and Christine Tobler, A literature survey of low-rank tensor approximation techniques, arXiv preprint arXiv:1302.7121, (2013), pp. 1–20.

[25] P R Halmos and V S Sunder, Bounded integral operators on L2 spaces, Ergebnisse der Mathematik und ihrer Grenzgebiete, Springer-Verlag, 1978.

[26] A. Hammerstein, Über die Entwicklung des Kernes linearer Integralgleichungen nach Eigenfunktionen, Sitzungsberichte Preuss. Akad. Wiss., (1923), pp. 181–184.

[27] G. H. Hardy and J. E. Littlewood, Some new properties of fourier constants, Mathematische Annalen, 97 (1927), pp. 159–209.

[28] R.A. Harshman, Foundations of the PARAFAC procedure: models and conditions for an'' explanatory'' multi-modal factor analysis, UCLA Working Papers in Phonetics, (1970), pp. 1–84.

[29] Jan S. Hesthaven and Tim Warburton, Nodal Discontinuous Galerkin Methods, vol. 54 of Texts in Applied Mathematics, Springer New York, New York, NY, 2008.

[30] K Jørgensen, Linear integral operators, Surveys and reference works in mathematics, Pitman Advanced Pub. Program, 1982.

[31] BN Khoromskij, O (dlog N)-Quantics Approximation of Nd Tensors in High-Dimensional Numerical Modeling, Constructive Approximation, (2011), pp. 257–280.

[32] B.N. Khoromskij and I. Oseledets, Quan-ter-Quant Collocation Approximation of Parameter-Dependent and Stochastic Elliptic PDEs, Computational Methods in Applied Mathematics, 10 (2010), pp. 376–394.

[33] BN Khoromskij and Christoph Schwab, Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs, SIAM Journal on Scientific Computing, 33 (2011), pp. 364–385.

[34] Tamara G. Kolda and Brett W. Bader, Tensor Decompositions and Applications, SIAM Review, 51 (2009), pp. 455–500.

[35] David A. Kopriva, Implementing Spectral Methods for Partial Differential Equations, Scientific Computation, Springer Netherlands, Dordrecht, 2009.

[36] E Kreyszig, Introductory functional analysis with applications, Wiley classics library, John Wiley & Sons, 2007.
SPECTRAL TENSOR-TRAIN DECOMPOSITION

[37] JB Kruskal, RA Harshman, and ME Lundy, How 3-MFA data can cause degenerate PARAFAC solutions, among other relationships, Multiway data analysis, (1989).

[38] Alexander Litvinenko, Hermann G Matthies, and Tarek A El-Moselhy, Sampling and Low-Rank Tensor Approximation of the Response Surface, in Monte Carlo and Quasi-Monte Carlo Methods 2012, Josef Dick, Frances Y. Kuo, Gareth W. Peters, and Ian H. Sloan, eds., vol. 65 of Springer Proceedings in Mathematics & Statistics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 535–551.

[39] M. Loève, Probability Theory, vol. I-II, Springer-Verlag, New York, 4 ed., 1978.

[40] Carla D Moravitz Martin and Charles F. Van Loan, A Jacobi-Type Method for Computing Orthogonal Tensor Decompositions, SIAM Journal on Matrix Analysis and Applications, 30 (2008), pp. 1219–1232.

[41] Hermann G. Matthies and Elmar Zander, Solving stochastic systems with low-rank tensor compression, Linear Algebra and its Applications, 436 (2012), pp. 3819–3838.

[42] Fabio Nobile, Raul Tempone, and C. G. Webster, An Anisotropic Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data, SIAM Journal on Numerical Analysis, 46 (2008), pp. 2411–2442.

[43] Anthony Nouy, Proper Generalized Decompositions and Separated Representations for the Numerical Solution of High Dimensional Stochastic Problems, Archives of Computational Methods in Engineering, 17 (2010), pp. 403–434.

[44] Anthony Nouy and Olivier P. Le Maître, Generalized spectral decomposition for stochastic nonlinear problems, Journal of Computational Physics, 228 (2009), pp. 202–235.

[45] IV Oseledets, Tensor-train decomposition, SIAM Journal on Scientific Computing, 33 (2011), pp. 2295–2317.

[46] Ivan Oseledets and Eugene Tyrtyshnikov, TT-cross approximation for multidimensional arrays, Linear Algebra and its Applications, 432 (2010), pp. 70–88.

[47] I V Oseledets, Approximation of 2 times 2d Matrices Using Tensor Decomposition, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 2130–2145.

[48] Dmitry Savostyanov and Ivan Oseledets, Fast adaptive interpolation of multi-dimensional arrays in tensor train format, The 2011 International Workshop on Multidimensional (nD) Systems, (2011), pp. 1–8.

[49] E Schmidt, Zur Theorie der linearen und nicht linearen Integralgleichungen Zweite Abhandlung, Mathematische Annalen, 63 (1907), pp. 433–476.

[50] Christoph Schwab and Radu Alexandru Todor, Karhunen-Loève approximation of random fields by generalized fast multipole methods, Journal of Computational Physics, 217 (2006), pp. 100–122.

[51] ND Sidiropoulos and Rasmus Bro, Tensor rank and the ill-posedness of the best low-rank approximation problem, SIAM Journal on Matrix Analysis and Applications, (2008), pp. 1–44.

[52] F Smithies, The eigen-values and singular values of integral equations, Proceedings of the London Mathematical Society, (1937).

[53] SA Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Dokl. Akad. Nauk SSSR, (1963).

[54] L Tamellini, O. Le Maître, and A Nouy, Model Reduction Based on Proper Generalized Decomposition for the Stochastic Steady Incompressible Navier–Stokes Equations, SIAM Journal on Scientific Computing, 36 (2014), pp. A1089–A1117.

[55] A Townsend and LN Trefethen, Continuous analogues of matrix factorizations, people.maths.ox.ac.uk, (2013), pp. 1–22.

[56] L N Trefethen and D Bau III, Numerical linear algebra, vol. 12, Society for Industrial and Applied Mathematics, 1997.

[57] LR Tucker, Implications of factor analysis of three-way matrices for measurement of change, in Problems in measuring change, C W Harris, ed., University of Wisconsin Press, Madison WI, 1963, pp. 122–137.

[58] Žimba, The bestL 2-approximation by finite sums of functions with separable variables, equationes mathematicae, 43 (1992), pp. 248–263.

[59] SR White, Density-matrix algorithms for quantum renormalization groups, Physical Review B, 48 (1993), pp. 345–356.

[60] Dongbin Xiu and Jan S. Hesthaven, High-Order Collocation Methods for Differential Equations with Random Inputs, SIAM Journal on Scientific Computing, 27 (2005), pp. 1118–1139.

[61] Zheng Zhang, Xiu Yang, Ivan V. Oseledets, George Em Karniadakis, and Luca Daniel, Enabling High-Dimensional Hierarchical Uncertainty Quantification by ANOVA and Tensor-Train Decomposition, arXiv preprint arXiv:1407.3023, (2014), p. 13.