Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Tensorisation of Vectors and their Efficient Convolution

by

Wolfgang Hackbusch

Preprint no.: 48 2010

Tensorisation of Vectors and their Efficient Convolution

Wolfgang Hackbusch
Max-Planck-Institut *Mathematik in den Naturwissenschaften*
Inselstr. 22, D-04103 Leipzig

Abstract

In recent papers the tensorisation of vectors has been discussed. In principle, this is the isomorphic representation of an $\mathbb{R}^n$ vector as a tensor. Black-box tensor approximation methods can be used to reduce the data size of the tensor representation. In particular, if the vector corresponds to a grid function, the resulting data size can become much smaller than n, e.g., $O(\log n) \ll n$. In this article we discuss vector operations, in particular, the convolution of two vectors which are given via a sparse tensor representation. We want to obtain the result again in the tensor representation. Furthermore, the cost of the convolution algorithm should be related to the operands' data sizes.

While $\mathbb{R}^n$ vectors can be considered as grid values of function, we also apply the same procedure to univariate functions.

AMS Subject Classifications: 15A69, 15A99, 44A35, 65F99, 65T99
Key words: tensorisation, tensor representation, hierarchical tensor representation, convolution, matrix-vector multiplication

1 Introduction

Tensorisation is an interpretation of an usual $\mathbb{R}^n$ vector as a tensor. For this purpose, we shall introduce a tensor space $\mathbf{V}$ and an isomorphism

$$\Phi = \Phi_n : \mathbf{V} \to \mathbb{R}^n$$

in §2. Because of the isomorphic structure, we have $\dim(\mathbf{V}) = n$. On the side of tensors we shall introduce certain tensor representations (tensor formats, see §3). They allow a simple truncation procedure, i.e., a tensor $\mathbf{v}$ can be replaced by $\tilde{\mathbf{v}}_\varepsilon$ with a guaranteed error bound $\|\mathbf{v} - \tilde{\mathbf{v}}_\varepsilon\| \le \varepsilon$. Often, the data size $N(\tilde{\mathbf{v}}_\varepsilon)$ of $\tilde{\mathbf{v}}_\varepsilon$ is much smaller than n (see Example 3.1 and the analysis by Grasedyck [3]). As a consequence, the tensorisation together with the truncation yields a black-box compression method for vectors in $\mathbb{R}^n$. However, the truncation and its analysis is not the subject of this article.

Here, we consider operations between vectors. The crucial point is that the *computational work* of the operations should be *related to the data size* of the operands. Assuming a data size $\ll n$, the cost should also be much smaller than the operation in the standard $\mathbb{R}^n$ vector format.

The first example of an operation is the scalar product $\langle v, w\rangle$ and is thought as an exercise introducing the recursive concept of the hierarchical format. Having ε-approximations $\tilde{\mathbf{v}}_\varepsilon$ and $\tilde{\mathbf{w}}_\varepsilon$ in the tensor format with data size $N(\tilde{\mathbf{v}}_\varepsilon), N(\tilde{\mathbf{w}}_\varepsilon) \ll n$, we are interested in the computation of $\langle \Phi(\tilde{\mathbf{v}}_\varepsilon), \Phi(\tilde{\mathbf{w}}_\varepsilon)\rangle \approx \langle v, w\rangle$ with a computational effort related to $N(\tilde{\mathbf{v}}_\varepsilon)$, $N(\tilde{\mathbf{w}}_\varepsilon)$ rather than n. Details will be given in §4.

However, the main interest of this article concerns the convolution operation $u := v \star w$ with $u_i = \sum_k v_k w_{i-k}$. We shall show that the convolution procedure can be applied directly to the tensor approximations $\tilde{\mathbf{v}}_\varepsilon$ and $\tilde{\mathbf{w}}_\varepsilon$. The algorithm is developed in §5 and its cost is related to the data sizes $N(\tilde{\mathbf{v}}_\varepsilon)$, $N(\tilde{\mathbf{w}}_\varepsilon)$.

In §6 we mention that instead of $\mathbb{R}^n$ we can also treat (finite dimensional subspaces of) function spaces. Again, operations like the scalar product or convolution of functions can be performed directly in the tensor format (see §6).

In the end we mention some generalisations. In particular, vectors can be replaced by matrices. Here, we remark how the matrix-vector multiplication can be performed using the tensor format for both matrices and vectors.

2 Tensorisation

2.1 Isomorphism Φ_n between $\bigotimes_{j=1}^{d} \mathbb{K}^2$ and $\mathbb{K}^n$

In the following, $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ is the underlying field. We number the entries of vectors from $\mathbb{K}^n$ by $0 \leq i \leq n-1$, i.e., $v = (v_i)_{i=0}^{n-1} \in \mathbb{K}^n$.

Assume that n is even. Then the tensor space $\mathbb{K}^{n/2} \otimes \mathbb{K}^2$ can be defined. Tensors $\mathbf{v} \in \mathbb{K}^{n/2} \otimes \mathbb{K}^2$ have entries $\mathbf{v}_{i_1 i_2} \in \mathbb{K}$ for $0 \leq i_1 \leq n/2 - 1$, $0 \leq i_2 \leq 1$. Since $\mathbb{K}^n$ and $\mathbb{K}^{n/2} \otimes \mathbb{K}^2$ have identical dimension, they are isomorphic. A special isomorphism is given by

$$\mathbf{v}_{i_1 i_2} = v_{i_1 + i_2 * n/2} \qquad \text{for } 0 \leq i_1 \leq \frac{n}{2} - 1,\ 0 \leq i_2 \leq 1.$$

For even $n/2$, we can replace $\mathbb{K}^{n/2}$ by the isomorphic space $\mathbb{K}^{n/4} \otimes \mathbb{K}^2$ and obtain $\mathbb{K}^n \cong \mathbb{K}^{n/4} \otimes \mathbb{K}^2 \otimes \mathbb{K}^2$. Iterating this process, we obtain the isomorphism

$$\Phi_n : \ \mathbf{V} := \bigotimes_{j=1}^{d} \mathbb{K}^2 \to \mathbb{K}^n \qquad \text{for } n = 2^d, \tag{2.1}$$

$$\mathbf{v} \in \mathbf{V} \mapsto v = (v_k)_{k=0}^{n-1} \in \mathbb{K}^n \text{ with } v_k = \mathbf{v}_{i_1 i_2 \ldots i_d}, \text{ where } k = \sum_{j=1}^{d} i_j 2^{j-1},\ i_j \in \{0, 1\}.$$

The tensor $\mathbf{v} \in \mathbf{V}$ has entries with d indices $i_j \in \{0, 1\}$, which correspond to the dual representation of the integer $k \in \{0, \ldots, n-1\}$. Particular tensors are the *elementary tensors*

$$\mathbf{v} = \bigotimes_{j=1}^{d} v^{(j)} \qquad \text{with } v^{(j)} \in \mathbb{K}^2, \tag{2.2a}$$

whose entries are

$$\mathbf{v}_{i_1 i_2 \ldots i_d} = \prod_{j=1}^{d} v_{i_j}^{(j)} \qquad \text{for } i_j \in \{0, 1\}. \tag{2.2b}$$

The restriction of the exponential function $\exp(\omega x)$ ($\omega \in \mathbb{K}$) to the grid $\left\{\frac{k}{n} : 0 \leq k \leq n-1\right\}$ yields the vector $v \in \mathbb{K}^n$ with $v_k = \exp\left(\omega \frac{k}{n}\right)$, which is subject of the next remark.

Remark 2.1 *Assume $n = 2^d$. The isomorphism Φ_n maps $v = \left(\exp\left(\omega \frac{k}{n}\right)\right)_{k=0}^{n-1} \in \mathbb{K}^n$ into an elementary tensor (2.2a) with $v^{(j)} = \binom{1}{\exp(2^{j-1}\omega/n)} \in \mathbb{K}^2$.*

The remarkable fact is that the elementary tensor (2.2a) is described by d $\mathbb{K}^2$-vectors, hence by $2d$ numbers. Note that $d = \log_2(n) \ll n$.

2.2 $\mathbb{K}^n$ and ℓ_0

In the following, it is helpful to introduce the set

$$\ell_0 := \{(a_i)_{i \in \mathbb{N}_0} : a_i = 0 \text{ for almost all } i \in \mathbb{N}_0\}$$

of *infinite sequences* with only finitely many nonzero entries. The embedding of $\mathbb{K}^n$ into ℓ_0 is defined by

$$\lambda_n : \mathbb{K}^n \to \ell_0, \qquad v \in \mathbb{K}^n \mapsto a = (a_i)_{i \in \mathbb{N}_0} \in \ell_0 \quad \text{with } a_i := \begin{cases} v_i & \text{for } 0 \leq i \leq n-1, \\ 0 & \text{for } i \geq n. \end{cases}$$

We define the degree of $a \in \ell_0$ by $\deg(a) := \max\{i \in \mathbb{N}_0 : a_i \neq 0\}$. Obviously, λ_n maps $\mathbb{K}^n$ into sequences $a \in \ell_0$ with $\deg(a) \leq n-1$.

For convenience we suppress the notation λ_n and identify $\mathbb{K}^n$ with the subset of sequences of degree $\leq n-1$.

The shift operator S^m ($m \in \mathbb{Z}$) is defined in ℓ_0 via

$$b = S^m(a) \text{ has entries } b_i = \begin{cases} a_{i-m} & \text{if } m \leq i, \\ 0 & \text{otherwise.} \end{cases}$$

2.3 Tensor space $\bigotimes_{j=1}^d \ell_0$

The embedding of $\mathbb{K}^n$ into ℓ_0 (and, particularly, of $\mathbb{K}^2$ into ℓ_0) allows to embed $\bigotimes_{j=1}^d \mathbb{K}^2$ into $\bigotimes_{j=1}^d \ell_0$. The mapping $\Phi_n : \bigotimes_{j=1}^d \mathbb{K}^2 \to \ell_0$ can be extended to[1]

$$\Phi_n : \bigotimes_{j=1}^{d} \ell_0 \to \ell_0\,,$$
$$\bigotimes_{j=1}^{d} v^{(j)} \mapsto a = (a_i)_{i\in\mathbb{N}_0} \qquad \text{with } a_k = \sum_{\substack{i_1,\ldots,i_d\in\mathbb{N}_0 \\ k=\sum_{j=1}^{d} i_j 2^{j-1}}} \prod_{j=1}^{d} v_{i_j}^{(j)}.$$

Note that $\Phi_n : \bigotimes_{j=1}^d \ell_0 \to \ell_0$ is not injective. Only in the case of $v^{(j)} \in \mathbb{K}^2$, the indices i_j are restricted to $\{0,1\}$ and each integer k has exactly one representation $k = \sum_{j=1}^d i_j 2^{j-1}$. Then the definition coincides with (2.1). By means of Φ_n we define an equivalence relation in $\bigotimes_{j=1}^d \ell_0$ via

$$\mathbf{v} \sim \mathbf{w} \quad \text{if and only if } \Phi_n(\mathbf{v}) = \Phi_n(\mathbf{w}) \qquad (\mathbf{v},\mathbf{w} \in \bigotimes_{j=1}^{d} \ell_0).$$

If $\deg(\mathbf{v}) := \deg(\Phi_n(\mathbf{v})) \le n-1$, there is a unique $\hat{\mathbf{v}} \in \bigotimes_{j=1}^d \mathbb{K}^2 \subset \bigotimes_{j=1}^d \ell_0$ with $\mathbf{v} \sim \hat{\mathbf{v}}$.

The shift operator can be used together with the tensor product.

Lemma 2.2 $\Phi_n\left(\bigotimes_{j=1}^{d} S^{m_j} v^{(j)}\right) = S^m \Phi_n\left(\bigotimes_{j=1}^{d} v^{(j)}\right)$ *holds for* $m = \sum_{j=1}^{d} m_j 2^{j-1}$.

So far, the action of S is defined for vectors of ℓ_0 only. For tensors, we set

$$S^m \bigotimes_{j=1}^{d} v^{(j)} := \left(S^m v^{(1)}\right) \otimes \bigotimes_{j=2}^{d} v^{(j)}, \tag{2.3}$$

i.e., the shift applies to the first direction. Then the statement of Lemma 2.2 can be written as

$$\bigotimes_{j=1}^{d} S^{m_j} v^{(j)} \sim S^m \bigotimes_{j=1}^{d} v^{(j)} \qquad \text{with } m = \sum_{j=1}^{d} m_j 2^{j-1}.$$

Since $\Phi_n\left(\mathbf{v} \otimes \binom{1}{0}\right) = \Phi_n\left(\mathbf{v}\right)$ for any tensor $\mathbf{v} \in \bigotimes_{j=1}^d \ell_0$, one obtains the following results:

$$\left.\begin{array}{lcl} \mathbf{v} \otimes \binom{1}{0} & \sim & \mathbf{v} \\ \mathbf{v} \otimes \binom{0}{1} & \sim & S^{2^d}\mathbf{v} \end{array}\right\} \quad \text{for all } \mathbf{v} \in \bigotimes_{j=1}^{d} \ell_0. \tag{2.4}$$

3 Tensor representation

The *full representation* of $\mathbf{v} \in \mathbf{V}$ stores all entries $\mathbf{v}_{i_1 i_2 \ldots i_d}$. This requires[2] $n = 2^d$ data and is equivalent to the full representation of the original vector $v \in \mathbb{K}^n$. On the other hand, we have already mentioned that an elementary tensor (2.2a) has a rather low storage size (only logarithmic in n). This fact is exploited by the next representation.

[1] In order to define a (multi-)linear mapping on a tensor space, it is sufficient to define the action on elementary tensors.

[2] In general applications, the size is not 2^d but m^d with $m \gg 2$. Then, $n = m^d$ can easily exceed the available storage, i.e., the full representation is not realisable.

3.1 r-term representation

Any tensor is by definition a linear combination of elementary tensors. When we bound the number of terms by $r \in \mathbb{N}_0$, we obtain the r-term representation. Here, $\mathbf{v} \in \mathbf{V}$ is said to possess an r-term representation, if there are vectors $v^{(j,\nu)} \in \mathbb{K}^2$ ($1 \leq j \leq d$, $1 \leq \nu \leq r$) with[3]

$$\mathbf{v} = \sum_{\nu=1}^{r} \bigotimes_{j=1}^{d} v^{(j,\nu)}. \tag{3.1}$$

The number r in (3.1) is called the *representation rank* of $\mathbf{v}$, while *rank*$(\mathbf{v})$ is the smallest possible r in (3.1). The set of all tensors satisfying (3.1) is denoted by $\mathcal{R}_r$. The storage size of $\mathbf{v} \in \mathcal{R}_r$ is $2rd$. As long as r is of moderate size, $2rd$ may be much smaller than n.

The next example shows the role of an approximation.

Example 3.1 *For $n = 2^d$ set $v = \left(f(\frac{k+1}{n})\right)_{k=0}^{n-1} \in \mathbb{K}^n$ for the function $f(x) = 1/x$ in $(0, 1]$. For any $r \in \mathbb{N}$, there is an approximation $v_{(r)} \in \mathbb{K}^n$ such that $\mathbf{v}_{(r)} := \Phi_n(v_{(r)})$ belongs to $\mathcal{R}_r$ and satisfies the component-wise error estimate*

$$\left|v_k - v_{(r),k}\right| \leq C_1 n \exp(-C_2 r) \qquad \text{with } C_1, C_2 > 0.$$

Hence, for a given error bound $\varepsilon > 0$, the choice $r = O(\log(n) + \log(1/\varepsilon))$ is sufficient. The storage size of the tensor $\mathbf{v}_{(r)}$ is $O(\log^2(n) + \log(n)\log(1/\varepsilon))$.

For a proof one uses the exponential sum approximation $\sum_{\nu=1}^{r} \omega_\nu \exp(-\alpha_\nu x)$ of $1/x$ in $[1/n, 1]$ with the corresponding error bound (cf. Braess-Hackbusch [1]). Inserting the grid values $x = (k+1)/n$, Remark 2.1 can be applied to each term. Hence, the storage is $2rd$. Choosing $r = O(\log(n) + \log(1/\varepsilon))$, we get the result from above.

If an r-term representation with moderate r is possible, like in Example 3.1, its use can be recommended, since operations with elementary tensor are particularly simple. However, the r-term representation becomes costly, if r is too large. Furthermore, the truncation of $\mathbf{v} \in \mathcal{R}_r$ to some $\tilde{\mathbf{v}} \in \mathcal{R}_s$ with $s < r$ is possible, but not quite easy (cf. Espig [2]).

3.2 Hierarchical representation

3.2.1 General case

Let $\mathbf{v} \in \mathbf{V} = \bigotimes_{\delta=1}^{d} V_\delta$. The hierarchical structure is described by the so-called *dimension partition tree T*. This is a binary tree with the following properties:

(i) the root is the set $\{1, \ldots, d\}$, while all vertices are certain subsets of $\{1, \ldots, d\}$,
(ii) a vertex $\alpha \in T$ is a leaf if and only if $\#\alpha = 1$, i.e., if α is a singleton $\{\delta\}$ for some $\delta \in \{1, \ldots, d\}$,
(iii) any non-leaf vertex $\alpha \in T$ has two sons $\alpha', \alpha'' \in T$ with the disjoint union $\alpha = \alpha' \cup \alpha''$.

Each vertex $\alpha \in T$ is associated to vector spaces $U_\alpha \subset W_\alpha$, which satisfy:

(i) for $\alpha = \{\delta\}$ (i.e., α is a leaf), $U_{\{\delta\}} \subset W_{\{\delta\}} := V_\delta$,
(ii) for a non-leaf vertex $\alpha \in T$ with sons $\alpha', \alpha'' \in T$ there holds

$$U_\alpha \subset W_\alpha := U_{\alpha'} \otimes U_{\alpha''}, \tag{3.2}$$

(iii) $\mathbf{v} \in U_{\{1,\ldots,d\}}$.

The dimension of the subspaces U_α is denoted by

$$r_\alpha = \dim(U_\alpha).$$

Remark 3.2 *The standard value of $r_{\{1,\ldots,d\}}$ is 1 because $U_{\{1,\ldots,d\}} = \operatorname{span}\{\mathbf{v}\}$ is sufficient. Only, if we want to represent several tensors by the same hierarchical representation, $r_d > 1$ makes sense.*

[3]Since $v^{(j,\nu)} = 0$ is not excluded, this definition allows also sums of less than r terms.

For the numerical realisation we use orthonormal bases of U_α. Here, we have to distinguish the case of leaves and non-leaves. For leaves $\alpha = \{\delta\} \in T$, a basis $\left\{b_1^{(\alpha)}, \ldots, b_{r_\alpha}^{(\alpha)}\right\}$ of U_α is stored. For the standard case $V_\delta = \mathbb{K}^{n_\delta}$, a storage of size $n_\delta r_{\{\delta\}}$ is needed.

For non-leaves $\alpha \in T$, the basis $\left\{b_1^{(\alpha)}, \ldots, b_{r_\alpha}^{(\alpha)}\right\}$ of U_α is characterised indirectly. $W_\alpha := U_{\alpha'} \otimes U_{\alpha''}$ has the induced basis $\left\{b_i^{(\alpha')} \otimes b_j^{(\alpha'')} : 1 \le i \le r_{\alpha'}, 1 \le j \le r_{\alpha''}\right\}$. Hence, $b_k^{(\alpha)} \in U_\alpha \subset W_\alpha$ has a representation

$$b_k^{(\alpha)} = \sum_{i=1}^{r_{\alpha'}} \sum_{j=1}^{r_{\alpha''}} C_{ij}^{(\alpha,k)} b_i^{(\alpha')} \otimes b_j^{(\alpha'')}. \tag{3.3}$$

Only the small-size matrix $C^{(\alpha,k)} = \left(C_{ij}^{(\alpha,k)}\right)_{1 \le i \le r_{\alpha'}, 1 \le i \le r_{\alpha''}}$ is to be stored.

Finally, the tensor $\mathbf{v} \in U_{\{1,\ldots,d\}}$ is given by $\mathbf{v} = \sum_{k=1}^{r_\alpha} c_k^{(\alpha)} b_k^{(\alpha)}$ for $\alpha = \{1, \ldots, d\}$, requiring the storage of $c^{(\alpha)} = \left(c_k^{(\alpha)}\right)_{k=1}^{r_\alpha} \in \mathbb{K}^{r_\alpha}$ (concerning $r_\alpha = 1$ see Remark 3.2).

3.2.2 Particular case for the present application

For the purpose of tensorisation it makes sense to consider the tensor product in the sequence $\left(\ldots\left(\left(\mathbb{K}^2 \otimes \mathbb{K}^2\right) \otimes \mathbb{K}^2\right) \otimes \ldots\right) \otimes \mathbb{K}^2$. The corresponding tree T becomes a linear tree.[4] Its vertices are $\{1, \ldots, \delta\}$ $(2 \le \delta \le d)$ and the singletons $\{\delta\}$. In particular, each vertex $\alpha = \{1, \ldots, \delta\}$ has the sons $\alpha' = \{1, \ldots, \delta - 1\}$ and $\alpha'' = \{\delta\}$.

The dimension of $V_\delta = \mathbb{K}^2$ is already so small that we do not try to find smaller subspaces $U_{\{\delta\}} \subset V_\delta$, i.e., we set $U_{\{\delta\}} := V_\delta = \mathbb{K}^2$.

Because of the special situation, we write
(i) U_δ instead of U_α for vertices $\alpha = \{1, \ldots, \delta\} \in T$. Here, we note that (3.2) becomes

$$U_\delta \subset U_{\delta-1} \otimes \mathbb{K}^2 \qquad (2 \le \delta \le d). \tag{3.4}$$

(ii) r_δ instead of r_α $(\alpha = \{1, \ldots, \delta\})$ for the dimension of U_δ,
(iii) $b_i^{(\delta)}$ instead of $b_i^{(\alpha)}$ $(\alpha = \{1, \ldots, \delta\})$ for the basis vectors of U_δ,
(iv) $C^{(\delta,k)}$ instead of $C^{(\alpha,k)}$ $(\alpha = \{1, \ldots, \delta\})$ for the coefficient matrix from (3.3).

Because of $U_{\{\delta\}} = \mathbb{K}^2$, we can avoid the use of the symbol $U_{\{\delta\}}$. The basis $\{b_1^{(\{\delta\})}, b_2^{(\{\delta\})}\}$ of $U_{\{\delta\}} = \mathbb{K}^2$ is fixed independently of δ by

$$b_1 := \binom{1}{0}, \quad b_2 := \binom{0}{1}. \tag{3.5}$$

Hence, formula (3.3) becomes

$$b_k^{(\delta)} = \sum_{i=1}^{r_{\delta-1}} \sum_{j=1}^{2} C_{ij}^{(\delta,k)} b_i^{(\delta-1)} \otimes b_j \qquad \text{for } \delta = 2, \ldots, d, \tag{3.6}$$

where the starting values are $b_i^{(1)} = b_i$ $(i = 1, 2$; cf. (3.5)).

The data of the hierarchical representation of a tensor $\mathbf{v} = \sum_{i=1}^{r_d} c_i b_i^{(d)}$ with $c := (c_i)_{i=1}^{r_d}$ are

$$\left\{\left(C^{(\delta,k)}\right)_{1 \le k \le r_\delta, 2 \le \delta \le d}, c\right\}. \tag{3.7}$$

If $r_\delta \le r$ holds for all vertices $\{1, \ldots, \delta\} \in T$, the overall storage cost is

$$(d-1)\, r^2 + r.$$

Remark 3.3 *The (minimal) dimension r_δ of the subspaces U_δ can be described algebraically. Let $\alpha = \{1, \ldots, \delta\}$ and write the tensor space $\bigotimes_{i=1}^d V_i$ in the form $\left(\bigotimes_{i=1}^\delta V_i\right) \otimes \left(\bigotimes_{i=\delta+1}^d V_i\right)$. A tensor $\mathbf{v}$ can be regarded as a matrix with the entries $v_{p,q}$, where $p = (i_1, \ldots, i_\delta)$ and $q = (i_{\delta+1}, \ldots, i_d)$. Then r_δ is the rank of this matrix.*

[4] For this choice the hierarchical format coincides with the so-called tensor tree format of Oseledets-Tyrtyshnikov [6].

4 Scalar product

The computation of the scalar product is considered as a first example for the use of the different representations, before we describe the convolution.

4.1 Scalar product of elementary tensors

The vector $v \in \mathbb{K}^n$ and the tensor $\mathbf{v} = \Phi_n^{-1} v \in \mathbf{V} := \bigotimes_{j=1}^{d} \mathbb{K}^2$ have the same components, only re-arranged in another ordering. Therefore, it is obvious that the Euclidean norms are equal:

$$\|v\|^2 := \sum_{i=1}^{n} |v_i|^2 = \sum_{i_1=1}^{2} \cdots \sum_{i_d=1}^{2} |\mathbf{v}_{i_1 \ldots i_d}|^2 =: \|\mathbf{v}\|^2 .$$

A similar statement holds for the scalar product:

$$\langle v, w \rangle := \sum_{i=1}^{n} v_i \overline{w_i} = \sum_{i_1=1}^{2} \cdots \sum_{i_d=1}^{2} \mathbf{v}_{i_1 \ldots i_d} \overline{\mathbf{w}_{i_1 \ldots i_d}} =: \langle \mathbf{v}, \mathbf{w} \rangle .$$

The concrete computation depends on the format in which $\mathbf{v}$ and $\mathbf{w}$ are given. First, we consider elementary tensors: $\mathbf{v} = \bigotimes_{j=1}^{d} v^{(j)}$ and $\mathbf{w} = \bigotimes_{j=1}^{d} w^{(j)}$ for certain $v^{(j)}, w^{(j)} \in V_j = \mathbb{K}^2$. Then the scalar product can be computed for each factor separately:

$$\left\langle \bigotimes_{j=1}^{d} v^{(j)}, \bigotimes_{j=1}^{d} w^{(j)} \right\rangle = \prod_{j=1}^{d} \left\langle v^{(j)}, w^{(j)} \right\rangle ,$$

where the latter scalar product belongs to $\mathbb{K}^2$, i.e., $\left\langle v^{(j)}, w^{(j)} \right\rangle = v_1^{(j)} \overline{w_1^{(j)}} + v_2^{(j)} \overline{w_2^{(j)}}$. Hence, the computation requires $4d - 1$ arithmetical operations.

4.2 Scalar product of tensors given in r-term format

Now, the tensors have the form $\mathbf{v} = \sum_{\nu=1}^{r_v} \bigotimes_{j=1}^{d} v^{(j,\nu)} \in \mathcal{R}_{r_v}$ and $\mathbf{w} = \sum_{\nu=1}^{r_w} \bigotimes_{j=1}^{d} w^{(j,\nu)} \in \mathcal{R}_{r_w}$. Obviously, the scalar product is given by the double sum

$$\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{\nu=1}^{r_v} \sum_{\mu=1}^{r_w} \prod_{j=1}^{d} \left\langle v^{(j,\nu)}, w^{(j,\mu)} \right\rangle .$$

Hence, the computational work is $r_v r_w \, (4d - 1)$.

4.3 Scalar product of tensors given in hierarchical representation

Let $\mathbf{v}$ and $\mathbf{w}$ be represented by the respective hierarchical data $\left\{ \left(C'^{(\delta,k)} \right)_{1 \le k \le r'_\delta, 2 \le \delta \le d}, c' \right\}$ and $\left\{ \left(C''^{(\delta,k)} \right)_{1 \le k \le r''_\delta, 2 \le \delta \le d}, c'' \right\}$ (cf. (3.7), i.e.,

$$\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{k=1}^{r'_d} \sum_{\ell=1}^{r''_d} c'_k \overline{c''_\ell} \left\langle b_k'^{(d)}, b_\ell''^{(d)} \right\rangle , \tag{4.1}$$

where $b_k'^{(d)}$ $[b_\ell''^{(d)}]$ denotes the basis vector associated to $\mathbf{v} \in U_d'$ $[\mathbf{w} \in U_d'']$. Next, we use the recursive definition of $b_k'^{(\delta)}, b_\ell''^{(\delta)}$ and the fact that b_m are the fixed orthonormal vectors from (3.5):

$$\begin{aligned}\left\langle b_k'^{(\delta)}, b_\ell''^{(\delta)} \right\rangle &= \left\langle \sum_{\nu=1}^{r'_{\delta-1}} \sum_{m=1}^{2} C'^{(\delta,k)}_{\nu m} b_\nu'^{(\delta-1)} \otimes b_m, \sum_{\mu=1}^{r''_{\delta-1}} \sum_{m=1}^{2} C''^{(\delta,\ell)}_{\mu m} b_\mu''^{(\delta-1)} \otimes b_m \right\rangle \\ &= \sum_{i=1}^{r'_{\delta-1}} \sum_{j=1}^{r''_{\delta-1}} \sum_{m=1}^{2} C'^{(\delta,k)}_{im} \overline{C''^{(\delta,\ell)}_{jm}} \left\langle b_i'^{(\delta-1)}, b_j''^{(\delta-1)} \right\rangle .\end{aligned}$$

Using $\left\langle b_i'^{(\delta-1)}, b_j''^{(\delta-1)} \right\rangle$ as entries of the matrix $B^{(\delta-1)} \in \mathbb{K}^{r'_{\delta-1} \times r''_{\delta-1}}$, we obtain from the previous line that

$$B^{(\delta)}_{k,\ell} = \left\langle b_k'^{(\delta)}, b_\ell''^{(\delta)} \right\rangle = \left\langle C^{(\delta,k,\ell)}, B^{(\delta-1)} \right\rangle_{\mathsf{F}} \qquad \text{with } C^{(\delta,k,\ell)} := C'^{(\delta,k)} \left(C''^{(\delta,\ell)} \right)^{\mathsf{H}}, \tag{4.2}$$

where $\langle A, B \rangle_{\mathsf{F}} := \sum_{i,j} A_{ij} \overline{B_{ij}}$ is the Frobenius scalar product of matrices. The dominant part of the computational cost is the building of the matrices $C^{(\delta,k,\ell)}$ for all k, ℓ, which requires $3r'_{\delta-1} r''_{\delta-1} r'_\delta r''_\delta$ operations.

By (4.2), $B^{(\delta)}$ can be obtained from $B^{(\delta-1)}$. This recursion starts with $B^{(1)} = B^{(\{1\})} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ and $r'_1 = r''_1 = 2$. Hence, the computation of $B^{(d)}$ by this recursion takes $3 \sum_{\delta=2}^{d} r'_{\delta-1} r''_{\delta-1} r'_\delta r''_\delta$ operations. If $r'_\delta, r''_\delta \le r$ for all δ, the asymptotic cost is $3\,(d-1)\,r^4$. Finally, $\langle \mathbf{v}, \mathbf{w} \rangle$ is obtained by (4.1), which is trivial because of the standard value $r'_d = r''_d = 1$ (cf. Remark 3.2).

5 Convolution

When we perform the convolution say of functions in d variables $x_1, \ldots, x_d$, it is well-known that for elementary tensors we may perform d *one*-dimensional convolutions instead:

$$\left(\prod_{j=1}^{d} f_j(x_j) \right) \star \left(\prod_{j=1}^{d} g_j(x_j) \right) = \prod_{j=1}^{d} (f_j \star g_j)(x_j).$$

The question arises, whether the convolution of standard vectors $v, w \in \mathbb{K}^n$ is equivalent to the *separate* convolution in each of the d directions of the artificially constructed tensorisations. The answer will be yes and no. After a direction-wise convolution we have to apply a certain carry-over procedure.

5.1 Convolution in ℓ_0 and $\mathbb{K}^n$

For the convolution of vectors $v, w \in \mathbb{K}^n$, we consider $\mathbb{K}^n$ as embedded in ℓ_0. The convolution in ℓ_0 is given by

$$a, b \in \ell_0 \quad \mapsto \quad c := a \star b \in \ell_0 \quad \text{with } c_k := \sum_{j=0}^{k} a_j b_{k-j}.$$

Remark 5.1 *a) If $v, w \in \mathbb{K}^n$, their convolution yields $v \star w \in \ell_0$ with $\deg(v \star w) \le 2n-2$. Hence, we may write $u := v \star w \in \mathbb{K}^{2n-1}$.*

b) When we represent the vectors $v, w \in \mathbb{K}^n$ by tensors from $\bigotimes_{\delta=1}^{d} \mathbb{K}^2$, the result $u := v \star w$ must be represented in $\bigotimes_{\delta=1}^{d+1} \mathbb{K}^2$ with d replaced by $d+1$ (the last entry of $u \in \mathbb{K}^{2n} = \mathbb{K}^{2^{d+1}}$ is $u_{2n-1} = 0$, since $\deg(u) \le 2n-2$).

The vector space ℓ_0 is isomorphic to the vector space $\mathbb{P}$ of *polynomials* of finite, but arbitrary degree:

$$\pi : \ell_0 \to \mathbb{P} \text{ with } \pi(a) = \sum_{j=0}^{\infty} a_j x^j . \tag{5.1}$$

According to the embedding $\mathbb{K}^n \hookrightarrow \ell_0$, we also use π as mapping from $\mathbb{K}^n$ into $\mathbb{P}$ (onto all polynomials of degree $\leq n-1$).

The convolution in ℓ_0 corresponds to the (pointwise) multiplication in $\mathbb{P}$:

$$c := a \star b \in \ell_0 \quad \Leftrightarrow \quad \pi(c) = \pi(a)\pi(b). \tag{5.2}$$

5.2 $\bigotimes_{\delta=1}^{d} \mathbb{K}^2$, $\bigotimes_{\delta=1}^{d} \ell_0$, and polynomials

The isomorphism π from (5.1) together with the embeddings $\mathbb{K}^n \hookrightarrow \ell_0$ and $\mathbb{K}^2 \hookrightarrow \ell_0$ allows another interpretation of tensors in $\bigotimes_{\delta=1}^{d} \mathbb{K}^2$. Consider an elementary tensor $\mathbf{v} = \bigotimes_{\delta=1}^{d} v^{(\delta)}$ ($v^{(\delta)} \in \mathbb{K}^2$) and the corresponding vector $v = \Phi_n(\mathbf{v}) \in \mathbb{K}^n$. Applying π to v, we obtain the polynomial $p := \pi(v)$ with $p(x) := \sum_{j=0}^{n-1} v_j x^j$.

For ease of notation, we shall write π instead of $\pi \circ \Phi_n$, i.e., $\pi : \bigotimes_{\delta=1}^{d} \mathbb{K}^2 \to \mathbb{P}$. The connection between $\mathbf{v} = \bigotimes_{\delta=1}^{d} v^{(\delta)}$ and $v = \Phi_n(\mathbf{v})$ on the side of polynomials is given by

$$p(x) = \prod_{\delta=1}^{d} p_\delta(x^{2^{\delta-1}}) \qquad \text{with } p := \pi(v) = \pi(\mathbf{v}),\ p_\delta := \pi(v^{(\delta)}), \tag{5.3}$$

where the linear polynomials $p_\delta(\xi) = v_0^{(\delta)} + v_1^{(\delta)}\xi$ are substituted by $\xi = x^{2^{\delta-1}}$. Note that the mapping $\pi : \bigotimes_{\delta=1}^{d} \mathbb{K}^2 \to \mathbb{P}$ is injective. Moreover, for any polynomial p of degree $\leq n-1$ ($n = 2^d$) there is a unique tensor $\mathbf{v} \in \bigotimes_{\delta=1}^{d} \mathbb{K}^2$ with $\pi(\mathbf{v}) = p$.

The mapping (5.3) can be extended from $\bigotimes_{\delta=1}^{d} \mathbb{K}^2$ to $\bigotimes_{\delta=1}^{d} v^{(\delta)} \in \bigotimes_{\delta=1}^{d} \ell_0$. The only difference is that now $p_\delta := \pi(v^{(\delta)})$ with $v^{(\delta)} \in \ell_0$ is a polynomial of arbitrary degree, and that $\pi : \bigotimes_{\delta=1}^{d} \ell_0 \to \mathbb{P}$ is no more injective. We note that $\mathbf{v} \sim \mathbf{w}$ if and only if $\pi(\mathbf{v}) = \pi(\mathbf{w})$ for $\mathbf{v}, \mathbf{w} \in \bigotimes_{\delta=1}^{d} \ell_0$.

5.3 Convolution of tensors

The elementary tensors $\mathbf{v} = \bigotimes_{\delta=1}^{d} v^{(\delta)}$ and $\mathbf{w} = \bigotimes_{\delta=1}^{d} w^{(\delta)}$ ($v^{(\delta)}, w^{(\delta)} \in \mathbb{K}^2$) represent the vectors $v = \Phi_n(\mathbf{v})$ and $w = \Phi_n(\mathbf{w})$ in $\mathbb{K}^n$. Therefore the convolution $\mathbf{v} \star \mathbf{w}$ must be defined such that

$$\Phi_n(\mathbf{v} \star \mathbf{w}) = \Phi_n(\mathbf{v}) \star \Phi_n(\mathbf{w}). \tag{5.4}$$

The right-hand side in (5.4) should not be used for the practical computation, since the vectors $\Phi_n(\mathbf{v})$ and $\Phi_n(\mathbf{w})$ have data size n and their convolution requires $O(n \log n)$ operations, whereas $\mathbf{v}$ and $\mathbf{w}$ have data size $O(d) = O(\log n)$ and the computational cost of $\mathbf{v} \star \mathbf{w}$ should be of similar size. Note that (5.4) defines only the equivalence class of $\mathbf{v} \star \mathbf{w}$.

It will turn out that the *term-wise* convolution is almost valid, i.e., $\mathbf{v} \star \mathbf{w} = \bigotimes_{\delta=1}^{d} \left(v^{(\delta)} \star w^{(\delta)}\right)$ holds, but its right-hand side is an element of $\bigotimes_{\delta=1}^{d} \ell_0$, not of $\bigotimes_{\delta=1}^{d} \mathbb{K}^2$, since $v^{(\delta)} \star w^{(\delta)}$ is a vector in $\mathbb{K}^3$ (cf. §5.1).

Lemma 5.2 *The convolution of $\mathbf{v} = \bigotimes_{\delta=1}^{d} v^{(\delta)}$ and $\mathbf{w} = \bigotimes_{\delta=1}^{d} w^{(\delta)}$ ($v^{(\delta)}, w^{(\delta)} \in \ell_0$) yields*

$$\bigotimes_{\delta=1}^{d} \left(v^{(\delta)} \star w^{(\delta)}\right), \qquad \textit{where } v^{(\delta)} \star w^{(\delta)} \in \ell_0.$$

Proof. Set $p := \pi(v)$, $p_\delta := \pi(v^{(\delta)})$ and $q := \pi(w)$, $q_\delta := \pi(w^{(\delta)})$. By definition, $\bigotimes_{\delta=1}^{d} \left(v^{(\delta)} \star w^{(\delta)}\right)$ corresponds to the vector u associated with the polynomial

$$\begin{aligned}\pi(u)(x) &= \prod_{\delta=1}^{d} \pi(v^{(\delta)} \star w^{(\delta)})(x^{2^{\delta-1}}) \underset{(5.2)}{=} \prod_{\delta=1}^{d} \pi(v^{(\delta)})(x^{2^{\delta-1}}) \cdot \pi(w^{(\delta)})(x^{2^{\delta-1}}) = \prod_{\delta=1}^{d} p_\delta(x^{2^{\delta-1}}) q_\delta(x^{2^{\delta-1}}) \\ &= \left(\prod_{\delta=1}^{d} p_\delta(x^{2^{\delta-1}})\right)\left(\prod_{\delta=1}^{d} q_\delta(x^{2^{\delta-1}})\right) = p(x)q(x) = \pi(v \star w)(x),\end{aligned}$$

which proves $u = v \star w$. ■

We recall Remark 5.1b: If $\mathbf{v}, \mathbf{w} \in \bigotimes_{\delta=1}^{d} \mathbb{K}^2$, the result is a tensor $\mathbf{u} := \mathbf{v} \star \mathbf{w}$ in $\bigotimes_{\delta=1}^{d+1} \mathbb{K}^2$. Lemma 5.3 describes the start at $d = 1$, while Lemma 5.4 can be used for the recursion.

Lemma 5.3 *The convolution of $v = \binom{\alpha}{\beta}, w = \binom{\gamma}{\delta} \in \mathbb{K}^2 = \bigotimes_{j=1}^{1} \mathbb{K}^2$ yields*

$$\binom{\alpha}{\beta} \star \binom{\gamma}{\delta} = \begin{pmatrix} \alpha\gamma \\ \alpha\delta + \beta\gamma \\ \beta\delta \\ 0 \end{pmatrix} = \binom{\alpha\gamma}{\alpha\delta+\beta\gamma} + S^2 \binom{\beta\delta}{0} \tag{5.5a}$$

$$= \Phi_4(\mathbf{v}) \quad \text{with } \mathbf{v} := \binom{\alpha\gamma}{\alpha\delta+\beta\gamma} \otimes \binom{1}{0} + \binom{\beta\delta}{0} \otimes \binom{0}{1} \in \bigotimes_{j=1}^{2} \mathbb{K}^2.$$

Furthermore, the shifted vector $S^1 \left(\binom{\alpha}{\beta} \star \binom{\gamma}{\delta} \right)$ has the tensor representation

$$S^1 \left(\binom{\alpha}{\beta} \star \binom{\gamma}{\delta} \right) = \begin{pmatrix} 0 \\ \alpha\gamma \\ \alpha\delta + \beta\gamma \\ \beta\delta \end{pmatrix} = \Phi_4(\mathbf{v}) \quad \text{with } \mathbf{v} := \binom{0}{\alpha\gamma} \otimes \binom{1}{0} + \binom{\alpha\delta+\beta\gamma}{\beta\delta} \otimes \binom{0}{1} \in \bigotimes_{j=1}^{2} \mathbb{K}^2. \tag{5.5b}$$

The basic identity is given in the next lemma.

Lemma 5.4 *Assume $\mathbf{v}, \mathbf{w} \in \bigotimes_{j=1}^{\delta-1} \mathbb{K}^2$ and $x = \binom{\alpha}{\beta}, y = \binom{\gamma}{\delta} \in \mathbb{K}^2$. Let the convolution result of $\mathbf{v}, \mathbf{w}$ be*

$$\mathbf{v} \star \mathbf{w} \sim \mathbf{a} = \mathbf{a}' \otimes \binom{1}{0} + \mathbf{a}'' \otimes \binom{0}{1} \in \bigotimes_{j=1}^{\delta} \mathbb{K}^2. \tag{5.6a}$$

Then, convolution of the tensors $\mathbf{v} \otimes x$ and $\mathbf{w} \otimes y$ yields

$$\begin{aligned} (\mathbf{v} \otimes x) \star (\mathbf{w} \otimes y) \sim \mathbf{u} = \mathbf{u}' \otimes \binom{1}{0} + \mathbf{u}'' \otimes \binom{0}{1} \in \bigotimes_{j=1}^{\delta+1} \mathbb{K}^2 \\ \text{with } \mathbf{u}' = \mathbf{a}' \otimes \binom{\alpha\gamma}{\alpha\delta+\beta\gamma} + \mathbf{a}'' \otimes \binom{0}{\alpha\gamma} \in \bigotimes_{j=1}^{\delta} \mathbb{K}^2 \\ \text{and } \mathbf{u}'' = \mathbf{a}' \otimes \binom{\beta\delta}{0} + \mathbf{a}'' \otimes \binom{\alpha\delta+\beta\gamma}{\beta\delta} \in \bigotimes_{j=1}^{\delta} \mathbb{K}^2. \end{aligned} \tag{5.6b}$$

Proof. Lemma 5.2 implies that

$$(\mathbf{v} \otimes x) \star (\mathbf{w} \otimes y) \sim (\mathbf{v} \star \mathbf{w}) \otimes z \qquad \text{with } z := x \star y \in \mathbb{K}^3 \subset \ell_0.$$

Assumption (5.6a) together with Lemma 2.2 and (2.4) yields

$$(\mathbf{v} \star \mathbf{w}) \otimes z \sim \left(\mathbf{a}' + S^{2^{\delta-1}} \mathbf{a}'' \right) \otimes z.$$

Again, Lemma 2.2 shows that

$$S^{2^{\delta-1}} \mathbf{a}'' \otimes z = S^{2^{\delta-1}} (\mathbf{a}'' \otimes z) \sim \mathbf{a}'' \otimes (Sz).$$

Using (5.5a,b), we obtain

$$\begin{aligned} \mathbf{a}' \otimes z \sim \mathbf{a}' \otimes \binom{\alpha\gamma}{\alpha\delta+\beta\gamma} \otimes \binom{1}{0} + \mathbf{a}' \otimes \binom{\beta\delta}{0} \otimes \binom{0}{1}, \\ \left(S^{2^{\delta-1}} \mathbf{a}'' \right) \otimes z \sim \mathbf{a}'' \otimes (Sz) \sim \mathbf{a}'' \otimes \binom{0}{\alpha\gamma} \otimes \binom{1}{0} + \mathbf{a}'' \otimes \binom{\alpha\delta+\beta\gamma}{\beta\delta} \otimes \binom{0}{1}. \end{aligned}$$

Summation of both identities yields the assertion of the lemma. ■

Corollary 5.5 $x, y \in \left\{ \binom{1}{0}, \binom{0}{1} \right\}$ *implies* $\binom{\alpha\gamma}{\alpha\delta+\beta\gamma}, \binom{0}{\alpha\gamma}, \binom{\beta\delta}{0}, \binom{\alpha\delta+\beta\gamma}{\beta\delta} \in \left\{ \binom{0}{0}, \binom{1}{0}, \binom{0}{1} \right\}$ *in (5.6b).*

Lemma 5.3 proves assumption (5.6a) for $\delta = 2$, while 5.4 shows that $\mathbf{v} \otimes x$ **and** $\mathbf{w} \otimes y$ satisfy requirement (5.6a) (for $\delta + 1$ instead of δ).

5.4 Convolution of elementary tensors or tensors in r-term format

Unfortunately, the convolution of elementary tensors does not result again in an elementary tensor. This is seen in (5.6b): even if $\mathbf{a}'$ and $\mathbf{a}''$ are elementary tensors, $\mathbf{u}'$ and $\mathbf{u}''$ are not. Instead, (5.6b) yields a sum of 2^δ terms.

As a consequence, the convolution of tensors in r-term format does not yield an s-term tensor with moderate representation rank s. Instead, one should convert tensors from the r-term format into the hierarchical format and apply the procedure of §5.5. We mention that tensors $\mathbf{v} \in \mathcal{R}_r$ allow a hierarchical representation with dimensions $r_\delta \leq r$ (cf. [5]).

5.5 Convolution of tensors in hierarchical format

We recall that the hierarchical format is characterised by the subspaces $U_\delta \subset \bigotimes_{j=1}^{\delta} \mathbb{K}^2$ satisfying (3.4): $U_\delta \subset U_{\delta-1} \otimes \mathbb{K}^2$. The essential observation is that also the results of the convolution yield subspaces with this property.

Note that there are *three* different tensors $\mathbf{v}$, $\mathbf{w}$, and $\mathbf{u} := \mathbf{v} * \mathbf{w}$ involving three different formats with three different subspace families U'_δ, U''_δ, and U_δ $(1 \leq \delta \leq d)$. The bases spanning these subspaces consist of the vectors $b_i^{\prime(\delta)}$, $b_i^{\prime\prime(\delta)}$, and $b_i^{(\delta)}$. The dimensions of the subspaces are r'_δ, r''_δ, and r_δ.

In order to map a tensor $\mathbf{a} = \mathbf{a}' \otimes \binom{1}{0} + \mathbf{a}'' \otimes \binom{0}{1}$ into $\mathbf{a}'$ and $\mathbf{a}''$, we introduce the mappings φ'_δ, φ''_δ with $\mathbf{a}' = \varphi'_\delta(\mathbf{a})$, $\mathbf{a}'' = \varphi''_\delta(\mathbf{a})$:

$$\varphi'_\delta, \varphi''_\delta : \bigotimes_{j=1}^{\delta} \mathbb{K}^2 \to \bigotimes_{j=1}^{\delta-1} \mathbb{K}^2 \qquad \text{with}$$
$$(\varphi'_\delta(\mathbf{v}))_{i_1 i_2 \ldots i_{\delta-1}} := \mathbf{v}_{i_1 i_2 \ldots i_{\delta-1},0} \text{ and } (\varphi''_\delta(\mathbf{v}))_{i_1 i_2 \ldots i_{\delta-1}} := \mathbf{v}_{i_1 i_2 \ldots i_{\delta-1},1} \quad (0 \leq i_j \leq 1)\,.$$

Theorem 5.6 *Let the tensors $\mathbf{v}, \mathbf{w} \in \bigotimes_{j=1}^{d} \mathbb{K}^2$ be represented by (possibly different) hierarchical formats using the respective subspaces U'_δ and U''_δ, $1 \leq \delta \leq d$, satisfying*

$$\begin{array}{lll} U'_1 = \mathbb{K}^2, & U'_\delta \subset U'_{\delta-1} \otimes \mathbb{K}^2, & \mathbf{v} \in U'_d\,, \\ U''_1 = \mathbb{K}^2, & U''_\delta \subset U''_{\delta-1} \otimes \mathbb{K}^2, & \mathbf{w} \in U''_d\,. \end{array} \tag{5.7a}$$

The subspaces

$$U_\delta := \operatorname{span}\{\varphi'_{\delta+1}(\mathbf{x} * \mathbf{y}), \varphi''_{\delta+1}(\mathbf{x} * \mathbf{y}) : \mathbf{x} \in U'_\delta,\, \mathbf{y} \in U''_\delta\} \qquad (1 \leq \delta \leq d) \tag{5.7b}$$

satisfy

$$U_1 = \mathbb{K}^2, \quad U_\delta \subset U_{\delta-1} \otimes \mathbb{K}^2, \quad \mathbf{v} * \mathbf{w} \in U_d. \tag{5.7c}$$

The dimension of U_δ can be bounded by

$$\dim(U_\delta) \leq 2 \dim(U'_\delta) \dim(U''_\delta).$$

Proof. 1) $U_1 = \mathbb{K}^2$ can be concluded from Lemma 5.3.

2) By assumption (5.7a), $\mathbf{x} \in U'_\delta \subset U'_{\delta-1} \otimes \mathbb{K}^2$ has a representation $\mathbf{x} = \mathbf{x}' \otimes \binom{1}{0} + \mathbf{x}'' \otimes \binom{0}{1}$ with $\mathbf{x}', \mathbf{x}'' \in U'_{\delta-1}$. The analogous statement holds for $\mathbf{y}$. Expansion of the sums yields $\mathbf{x} * \mathbf{y} = \big(\mathbf{x}' \otimes \binom{1}{0}\big) * \big(\mathbf{y}' \otimes \binom{1}{0}\big) + \ldots$ For each term, Lemma 5.4 states that $\varphi'_{\delta+1}\big(\big(\mathbf{x}' \otimes \binom{1}{0}\big) * \big(\mathbf{y}' \otimes \binom{1}{0}\big)\big) = \mathbf{u}'$ and $\varphi''_{\delta+1}(\ldots) = \mathbf{u}''$ belong to $U_{\delta-1} \otimes \mathbb{K}^2$ (cf. (5.6b)). Hence, $\varphi'_{\delta+1}(\mathbf{x} * \mathbf{y}), \varphi''_{\delta+1}(\mathbf{x} * \mathbf{y}) \in U_{\delta-1} \otimes \mathbb{K}^2$, and the definition of U_δ implies the inclusion $U_\delta \subset U_{\delta-1} \otimes \mathbb{K}^2$.

3) $\mathbf{v} \in U'_d$ and $\mathbf{w} \in U''_d$ together with the definition of U_d lead to $\mathbf{v} * \mathbf{w} \in U_d$.

4) The bound of $\dim(U_\delta)$ follows directly from (5.7b). ∎

For $\delta = 1, \ldots, d$, the numerical scheme has

1. to introduce an orthonormal basis $\{b_1^{(\delta)}, \ldots, b_{r_\delta}^{(\delta)}\}$ of U_δ, where $r_\delta := \dim(U_\delta)$, and

2. to represent the convolution $b_i^{\prime(\delta)} * b_j^{\prime\prime(\delta)}$ by

$$b_i^{\prime(\delta)} * b_j^{\prime\prime(\delta)} = \sum_{m=1}^{2} \sum_{k=1}^{r_\delta} \beta_{ij,km}^{(\delta)} b_k^{(\delta)} \otimes b_m. \tag{5.8}$$

As soon as the β-coefficients from (5.8) are known, general products $\mathbf{x} * \mathbf{y}$ of $\mathbf{x} \in U_\delta'$ and $\mathbf{y} \in U_\delta''$ can be evaluated easily as shown in the next remark.

Remark 5.7 *Let* $\mathbf{x} = \sum_{i=1}^{r_\delta'} \xi_i b_i^{\prime(\delta)} \in U_\delta'$ *and* $\mathbf{y} = \sum_{j=1}^{r_\delta''} \eta_j b_j^{\prime\prime(\delta)} \in U_\delta''$. *Then convolution yields*

$$\mathbf{x} * \mathbf{y} = \mathbf{z} = \mathbf{z}' \otimes \binom{1}{0} + \mathbf{z}'' \otimes \binom{0}{1} \quad \textit{with} \quad \mathbf{z}' = \sum_{k=1}^{r_\delta} \zeta_k' b_k^{(\delta)}, \; \mathbf{z}'' = \sum_{k=1}^{r_\delta} \zeta_k'' b_k^{(\delta)},$$

$$\textit{where} \quad \zeta_k' = \sum_{i=1}^{r_\delta'} \sum_{j=1}^{r_\delta''} \xi_i \eta_j \beta_{ij,k1}^{(\delta)} \quad \textit{and} \quad \zeta_k'' = \sum_{i=1}^{r_\delta'} \sum_{j=1}^{r_\delta''} \xi_i \eta_j \beta_{ij,k2}^{(\delta)}$$

with $\beta_{ij,km}^{(\delta)}$ *from (5.8). The computation of* ζ_k', ζ_k'' $(1 \le k \le r_\delta)$ *requires* $4 r_\delta r_\delta' (r_\delta'' + 1)$ *operations.*

Start $\delta = 1$. For $\delta = 1$, $U_1' = U_1'' = U_1 = \mathbb{K}^2$ holds, and the bases are identically given by $b_1^{\prime(1)} = b_1^{\prime\prime(1)} = b_1^{(1)} = \binom{1}{0}$ and $b_2^{\prime(1)} = \ldots = \binom{0}{1}$. The β-coefficients from (5.8) are

		$k=1$, $j=1$	$k=1$, $j=2$	$k=2$, $j=1$	$k=2$, $j=2$
$\beta_{ij,k1}^{(\delta)}$:	$i=1$	1	0	0	1
	$i=2$	0	0	1	0
$\beta_{ij,k2}^{(\delta)}$:	$i=1$	0	0	0	0
	$i=2$	0	1	0	0

Recursion step from $\delta - 1$ **to** $\delta \in \{2, \ldots, d\}$. By induction, the basis $\left\{ b_k^{(\delta-1)} : 1 \le k \le r_{\delta-1} \right\}$ is already available. In a first step, we represent $b_i^{\prime(\delta)} * b_j^{\prime\prime(\delta)}$ in the orthonormal basis $\{ b_\nu^{(\delta-1)} \otimes \binom{1}{0}, b_\nu^{(\delta-1)} \otimes \binom{0}{1} : 1 \le \nu \le r_{\delta-1} \}$, which spans $U_{\delta-1} \otimes \mathbb{K}^2$. For this purpose, we recall that $b_i^{\prime(\delta)}$ and $b_j^{\prime\prime(\delta)}$ have representations $b_i^{\prime(\delta)} = \sum_{\nu=1}^{r_{\delta-1}} \sum_{\mu=1}^{2} C_{\nu\mu}^{\prime(\delta,i)} b_\nu^{\prime(\delta-1)} \otimes b_\mu$ and $b_j^{\prime\prime(\delta)} = \sum_{\rho=1}^{r_{\delta-1}} \sum_{\sigma=1}^{2} C_{\rho\sigma}^{\prime\prime(\delta,j)} b_\rho^{\prime\prime(\delta-1)} \otimes b_\sigma$ (cf. (3.3)). The result

$$b_{ij}^{(\delta)} := b_i^{\prime(\delta)} * b_j^{\prime\prime(\delta)} = \sum_{\mu=1}^{2} \sum_{\sigma=1}^{2} \underbrace{\sum_{\nu=1}^{r_{\delta-1}'} \sum_{\rho=1}^{r_{\delta-1}''} C_{\nu\mu}^{\prime(\delta,i)} C_{\rho\sigma}^{\prime\prime(\delta,j)} \left(b_\nu^{\prime(\delta-1)} \otimes b_\mu \right) * \left(b_\rho^{\prime\prime(\delta-1)} \otimes b_\sigma \right)}_{=: b_{ij,\mu\sigma}^{(\delta)}} \tag{5.9}$$

is a sum of the four terms $b_{ij,\mu\sigma}^{(\delta)}$ $(1 \le \mu, \sigma \le 2)$. Each term can be split into

$$b_{ij,\mu\sigma}^{(\delta)} = b_{ij,\mu\sigma}^{\prime(\delta)} \otimes b_1 + b_{ij,\mu\sigma}^{\prime\prime(\delta)} \otimes b_2 \quad \text{with } b_{ij,\mu\sigma}^{\prime(\delta)} = \varphi_{\delta+1}'(b_{ij,\mu\sigma}^{(\delta)}) \text{ and } b_{ij,\mu\sigma}^{\prime\prime(\delta)} = \varphi_{\delta+1}''(b_{ij,\mu\sigma}^{(\delta)}). \tag{5.10}$$

As example, we consider the case $\mu = \sigma = 1$. Then $x = y = b_1$ leads to $\alpha = \gamma = 1$, $\beta = \delta = 0$ and to $\binom{\alpha\gamma}{\alpha\delta+\beta\gamma} = b_1, \binom{0}{\alpha\gamma} = b_2$ in Lemma 5.4. The product $\left(b_\nu^{\prime(\delta-1)} \otimes b_\mu \right) * \left(b_\rho^{\prime\prime(\delta-1)} \otimes b_\sigma \right)$ can be evaluated by (5.6a,b):

$$\begin{aligned} & \varphi_{\delta+1}' \left(\left(b_\nu^{\prime(\delta-1)} \otimes b_1 \right) * \left(b_\rho^{\prime\prime(\delta-1)} \otimes b_1 \right) \right) \\ \underset{(5.6b)}{=} \; & \varphi_\delta' \left(b_\nu^{\prime(\delta-1)} * b_\rho^{\prime\prime(\delta-1)} \right) \otimes b_1 + \varphi_\delta'' \left(b_\nu^{\prime(\delta-1)} * b_\rho^{\prime\prime(\delta-1)} \right) \otimes b_2 \\ \underset{(5.8)}{=} \; & \sum_{m=1}^{2} \sum_{k=1}^{r_{\delta-1}} \beta_{\nu\rho,km}^{(\delta-1)} b_k^{(\delta-1)} \otimes b_m \in U_{\delta-1} \otimes \mathbb{K}^2, \end{aligned}$$

while $\binom{\beta\delta}{0} = \binom{\alpha\delta+\beta\gamma}{\beta\delta} = 0$ implies

$$\varphi''_{\delta+1}\left(\left(b_\nu'^{(\delta-1)} \otimes b_1\right) * \left(b_\rho''^{(\delta-1)} \otimes b_1\right)\right) = 0.$$

This determines the components in (5.10) for $\mu = \sigma = 1$. Together with the definition of $b_{ij,\mu\sigma}^{(\delta)}$ in (5.9), we obtain $b_{ij,11}^{(\delta)}$ as a linear combination of the basis vectors $b_k'^{(\delta-1)} \otimes b_m$. Similar representations hold for the other $b_{ij,\mu\sigma}^{(\delta)}$ ($(\mu,\sigma) \neq (1,1)$). Finally, we have

$$b_{ij}^{(\delta)} := b_i'^{(\delta)} * b_j''^{(\delta)} = b_{ij}'^{(\delta)} \otimes b_1 + b_{ij}''^{(\delta)} \otimes b_2 \quad \text{with}$$
$$b_{ij}'^{(\delta)} = \sum_{k=1}^{r_{\delta-1}} \sum_{m=1}^{2} \gamma_{ij,km}'^{(\delta)} b_k^{(\delta-1)} \otimes b_m, \quad b_{ij}''^{(\delta)} = \sum_{k=1}^{r_{\delta-1}} \sum_{m=1}^{2} \gamma_{ij,km}''^{(\delta)} b_k^{(\delta-1)} \otimes b_m,$$

where the computation of the coefficients $\gamma_{ij,km}'^{(\delta)}$, $\gamma_{ij,km}''^{(\delta)}$ from $\beta_{\nu\rho,km}^{(\delta-1)}$, $C_{\nu\mu}'^{(\delta,i)}$, $C_{\rho\sigma}''^{(\delta,j)}$ requires $8r''_\delta r'_{\delta-1} r_{\delta-1}\left(r''_{\delta-1} + r'_\delta\right)$ operations.

The vectors $b_{ij}'^{(\delta)}$, $b_{ij}''^{(\delta)}$ span the subspace $U_\delta \subset U_{\delta-1} \otimes \mathbb{K}^2$. Next, we have find an orthonormal basis $\{b_\nu^{(\delta)} : 1 \leq \nu \leq r_\delta\}$ of U_δ, where $r_\delta := \dim(U_\delta)$. More precisely, we have to find the coefficients $C_{ij}^{(\delta,\nu)}$ such that $b_\nu^{(\delta)} = \sum_{i=1}^{r_{\delta-1}} \sum_{j=1}^{2} C_{ij}^{(\delta,k)} b_i^{(\delta-1)} \otimes b_j$ (cf. (3.6)). This concludes the induction step from $\delta - 1$ to δ.

The construction of the new orthonormal basis may use the QR decomposition. Here, we use the Gram matrix G_δ requiring the pairwise scalar products

$$\left\langle b_{ij}'^{(\delta)}, b_{\ell m}'^{(\delta)} \right\rangle, \ \left\langle b_{ij}'^{(\delta)}, b_{\ell m}''^{(\delta)} \right\rangle, \ \left\langle b_{ij}''^{(\delta)}, b_{\ell m}''^{(\delta)} \right\rangle \qquad \text{for } 1 \leq i, \ell \leq r'_\delta, \ 1 \leq j, m \leq r''_\delta.$$

The indices $q, p \in \{1, \ldots, 2r'_\delta r''_\delta\}$ of $G_\delta = (g_{pq})$ can be considered as an ordering of the triples $(ij\alpha)$ with $1 \leq i \leq r'_\delta$, $1 \leq j \leq r''_\delta$, $1 \leq \alpha \leq 2$ referring to $b_{ij1}^{(\delta)} := b_{ij}'^{(\delta)}$ and $b_{ij2}^{(\delta)} := b_{ij}''^{(\delta)}$. The computation of G_δ takes $8\left(r'_\delta r''_\delta\right)^2 r_{\delta-1}$ operations (note that the basis $b_k^{(\delta-1)} \otimes b_m$ $(1 \leq k \leq r_{\delta-1}, 1 \leq m \leq 2)$ is orthonormal so that, e.g., $\left\langle b_{ij}'^{(\delta)}, b_{\nu\mu}'^{(\delta)} \right\rangle = \sum_{k=1}^{r_{\delta-1}} \sum_{m=1}^{2} \gamma_{ij,km}'^{(\delta)} \overline{\gamma_{\nu\mu,km}'^{(\delta)}}$). The computation of the Cholesky decomposition $G_\delta = L_\delta L_\delta^{\mathsf{H}}$ requires $\frac{4}{3}\left(r'_\delta r''_\delta\right)^3$ operations. Here we assume that the indices are such that the lower triangular matrix L has the form

$$L = \begin{bmatrix} L_\delta & O \\ * & O \end{bmatrix} \qquad \text{with } L_\delta \in \mathbb{K}^{r_\delta \times r_\delta}, \ r_\delta = rank(G_\delta).$$

Set[5] $A := \begin{bmatrix} L_\delta^{-1} & O \end{bmatrix}$. Then $b_\nu^{(\delta)} := \sum a_{\nu,(ij\alpha)} b_{ij\alpha}^{(\delta)}$ for $1 \leq \nu \leq r_\delta$ represents the new orthonormal basis. The coefficients $C_{km}^{(\delta,k)}$ from (3.6) are obtained via

$$C_{km}^{(\delta,\nu)} = \sum_{i,j} \left(a_{\nu,(ij1)} \gamma_{ij,km}'^{(\delta)} + a_{\nu,(ij2)} \gamma_{ij,km}''^{(\delta)}\right) \qquad \text{for } 1 \leq \nu \leq r_\delta, \, 1 \leq k \leq r_{\delta-1}, \, 1 \leq m \leq 2.$$

The corresponding cost is $2r_{\delta-1} r_\delta^2$ (cf. Footnote 5). By construction, we obtain the coefficients $\beta_{ij,km}^{(\delta)}$ from (5.8). Note that (5.8) is equivalent to $b_{ij}'^{(\delta)} = \sum_{k=1}^{r_\delta} \beta_{ij,k1}^{(\delta)} b_k^{(\delta)}$ and $b_{ij}''^{(\delta)} = \sum_{k=1}^{r_\delta} \beta_{ij,k2}^{(\delta)} b_k^{(\delta)}$. This finishes the computation of $\beta_{\ldots}^{(\delta)}$ from $\beta_{\ldots}^{(\delta-1)}$.

Adding the above mentioned costs of the partial steps, we obtain the total cost of the recursion step from $\delta - 1$ to δ, which is to be summed over all δ:

$$8r''_\delta r'_{\delta-1} r_{\delta-1}\left(r''_{\delta-1} + r'_\delta\right) + 8\left(r'_\delta r''_\delta\right)^2 r_{\delta-1} + \frac{4}{3}\left(r'_\delta r''_\delta\right)^3 + 2r_{\delta-1} r_\delta^2 \qquad \text{for } 2 \leq \delta \leq d.$$

[5] The inverse is not computed explicitly. Instead, back substitution is used, when A is applied to a vector (matrix).

6 Function spaces

6.1 Tensorisation of functions

So far, we have considered vectors which can be considered as discrete grid functions. Now we discuss true functions and their convolution. So simplify the setting,[6] we consider piecewise continuous functions on the unit interval $[0,1]$. Let $n = 2^d$ be fixed and define $C_{pw}([0,1])$ by functions which are uniformly continuous on each subinterval $[k/n, (k+1)/n)$. Many approximation schemes uses a piecewise approximation of functions on the described subintervals (hp-method,[7] splines, wavelets etc.). In the latter case, the function restricted to the subinterval belongs to a certain (finite dimensional) subspace $V_0 \subset C([0,1/n])$ (polynomials, generating wavelet function, etc.).

We shall construct an isomorphism between the tensor space

$$C\left(\left[0,\tfrac{1}{n}\right]\right) \otimes \bigotimes_{j=1}^{d} \mathbb{K}^2$$

and $C_{pw}([0,1])$. Since we know already the isomorphism between $\bigotimes_{j=1}^{d} \mathbb{K}^2$ and $\mathbb{K}^n$, it is enough to describe the isomorphism

$$\Psi_n : \mathbf{V} := C\left(\left[0,\tfrac{1}{n}\right]\right) \otimes \mathbb{K}^n \to C_{pw}([0,1]).$$

A tensor $\mathbf{v} \in \mathbf{V}$ can be considered as a function on $[0,\frac{1}{n}] \times \{0,\ldots,n-1\}$. The first argument $x \in [0,\frac{1}{n}]$ is a continuous variable, whereas $i \in \{0,\ldots,n-1\}$ is discrete. The definition of Ψ_n is given via

$$f = \Psi_n(\mathbf{v}) \quad \text{with } f\left(\frac{k}{n} + x\right) = \mathbf{v}\left(x,k\right) \qquad \text{for all } 0 \le x < \frac{1}{n} \text{ and } 0 \le k \le n-1.$$

The continuous version of the embedding $\mathbb{K}^2 \hookrightarrow \mathbb{K}^n \hookrightarrow \ell_0$ from §2.2 is $C([0,\frac{1}{n}]) \hookrightarrow C_{pw}([0,1]) \hookrightarrow C_{pw,0}([0,\infty))$, where $f \in C_{pw,0}([0,\infty))$ are piecewise continuous functions with finite support. The convolution maps $C_{pw,0}([0,\infty)) \times C_{pw,0}([0,\infty))$ into $C_{pw,0}([0,\infty))$:

$$\left(f \star g\right)(x) = \int_0^x f(t)g(x-t)\mathrm{d}t \qquad \text{for } f,g \in C_{pw,0}([0,\infty)) \text{ and } 0 \le x < \infty.$$

As in §2.3, we embed $\mathbf{V} = C([0,\frac{1}{n}]) \otimes \mathbb{K}^n$ in $C_{pw,0}([0,\infty)) \otimes \ell_0$. The interpretation of $\mathbf{v} \in C_{pw,0}([0,\infty)) \otimes \ell_0$ is

$$f = \Psi_n(\mathbf{v}) \qquad \text{has the values } f(x) = \sum_{0 \le k \le nx} \mathbf{v}\left(x - \tfrac{k}{n}, k\right) \text{ for } x \ge 0.$$

Note that by definition of ℓ_0 the latter sum is finite. Again, we use the notation

$$\mathbf{v} \sim \mathbf{w} \quad \Leftrightarrow \quad \Psi_n(\mathbf{v}) = \Psi_n(\mathbf{w}).$$

The convolution $\mathbf{v} \star \mathbf{w}$ has to satisfy $\Psi_n\left(\mathbf{v} \star \mathbf{w}\right) = \Psi_n(\mathbf{v}) \star \Psi_n(\mathbf{w})$ (cf. (5.4)).

We introduce the shift operator $S : C_{pw,0}([0,\infty)) \to C_{pw,0}([0,\infty))$ via

$$\left(S^m f\right)(x) = \begin{cases} f\left(x - \frac{m}{n}\right) & \text{for } \frac{m}{n} \ge x \in [0,\infty), \\ 0 & \text{otherwise.} \end{cases}$$

Then, the isomorphism Ψ_n can be rewritten as

$$\Psi_n(\mathbf{v})(x) = \sum_{k=0}^{\infty} S^k \mathbf{v}\left(x,k\right). \tag{6.1a}$$

[6] Other functions spaces are possible as well. If the functions are not required to be continuous, a piecewise definition is not required. The advantage of $C_{pw}([0,1])$ is that the convolution result lies again in $C_{pw}([0,1])$.

[7] In [4] we have considered the convolution of piecewise polynomials on a refined grid. Such an hp approximation is a possible sparsification of data and requires a particular convolution algorithm. The tensorisation is even more general, since hp approximations allow a sparse representation by the tensor representation (cf. [3]) and can make use of even other types of functions.

Furthermore, the identity

$$\left(S^k f\right) \star \left(S^\ell g\right) = S^{k+\ell}\left(f \star g\right) \tag{6.1b}$$

holds.

The analogue of Lemma 5.2 is the following statement.

Lemma 6.1 *Let $\mathbf{v} = \varphi \otimes x$ and $\mathbf{w} = \psi \otimes y$ be elementary tensors from $C_{pw,0}([0,\infty)) \otimes \ell_0$. Then*

$$\mathbf{v} \star \mathbf{w} \sim (\varphi \star \psi) \otimes (x \star y).$$

Proof. By (6.1a) we have $f := \Psi_n(\mathbf{v}) = \sum_{k=0}^{\infty} x_k \left(S^k \varphi\right)$ and $g := \Psi_n(\mathbf{w}) = \sum_{\ell=0}^{\infty} y_\ell \left(S^\ell \psi\right)$. By (6.1b), convolution yields

$$f \star g = \sum_{k=0}^{\infty} \sum_{\ell=0}^{\infty} x_k y_\ell S^{k+\ell}\left(f \star g\right).$$

We set $z := x \star y$ (discrete convolution in ℓ_0) and substitute $m = k + \ell$. The identity

$$f \star g = \sum_{m=0}^{\infty} \sum_{k=0}^{m} x_k y_{m-k} S^m \left(f \star g\right) = \sum_{m=0}^{\infty} z_m S^m \left(f \star g\right) = \Psi_n\left((f \star g) \otimes z\right)$$

proves the assertion. ■

Remark 6.2 *If the support of $f, g \in C_{pw,0}([0,\infty))$ is contained in $[0, 1/n]$, $u := f \star g$ belongs to $C([0, 2/n])$ and can be written as $u = u' \otimes b_1 + u'' \otimes b_2$ with $u' := u|_{[0,1/n]}$ and $u'' := u(\cdot + 1/n)|_{[0,1/n]}$. Here, the unit vectors $b_1, b_2 \in \mathbb{K}^2$ from (3.5) are considered as embedded in ℓ_0.*

Now, we can replace $\mathbb{K}^n$ by $\bigotimes_{j=1}^{d} \mathbb{K}^2$ (and ℓ_0 by $\bigotimes_{j=1}^{d} \ell_0$). Hence, $C_{pw}([0,1])$ becomes isomorphic to $\mathbf{V} = \bigotimes_{j=0}^{d} V_j = C([0,1/n]) \otimes \bigotimes_{j=1}^{d} \mathbb{K}^2$.

6.2 Hierarchical representation

We use again the hierarchical representation from §3.2.2 with the following modifications:

1. The dimension index set is $\{0, 1, \ldots, d\}$ instead of $\{1, \ldots, d\}$. The vector space V_0 is the function space on $[0, 1/n]$, e.g., $C([0, 1/n])$, whereas all other spaces are $V_j = \mathbb{K}^2$ as before.
2. For $j = 0$ we have to specify a (finite dimensional) subspace $U_0 \subset V_0$ by means of a basis $\{b_1^{(0)}, \ldots, b_{r_0}^{(0)}\}$. If we want to use a piecewise polynomial approximation, $b_i^{(0)}$ may be the Legendre polynomials of degree $i - 1$ mapped onto $[0, 1/n]$. In the case of Example 3.1,[8] the basis functions are exponentials $b_\nu^{(0)}(x) = \exp(-\alpha_\nu x)$ for certain $\alpha_\nu > 0$, $1 \le \nu \le r_0$.

A tensor $\mathbf{v} \in \mathbf{V} = \bigotimes_{j=0}^{d} V_j$ is now characterised by the data

$$\left(\left(b_i^{(0)}\right)_{1 \le i \le r_0}, \left(C^{(\delta,k)}\right)_{1 \le k \le r_\delta, 1 \le \delta \le d}, c\right),$$

i.e., $\mathbf{v} = \sum_{i=1}^{r_d} c_i b_i^{(d)}$ holds, where the basis vectors $b_i^{(d)}$ are recursively defined via (3.6). Differently from before, (3.6) is also used for $\delta = 1$, and the basis $(b_i^{(0)})_{1 \le i \le r_0}$ of $U_0 \subset V_0$ must be prescribed explicitly (for $2 \le \delta \le d$, the basis of $U_\delta = V_\delta = \mathbb{K}^2$ is given by (3.5)).

[8] Here, we have to shift the function by $1/n$ to avoid the singularity: $f(x) = 1/(x + 1/n)$.

6.3 Scalar product

We discuss the scalar product of tensors $\mathbf{v}, \mathbf{w} \in \mathbf{V}$ given in the hierarchical format, i.e., $\mathbf{v}$ and $\mathbf{w}$ are given by the respective data $\left((b_i'^{(0)})_{1 \le i \le r_0'}, \left(C'^{(\delta,k)}\right)_{1\le k \le r_\delta', 1 \le \delta \le d}, c' \right)$ and $\left((b_i''^{(0)})_{1 \le i \le r_0''}, \left(C''^{(\delta,k)}\right)_{1\le k \le r_\delta'', 1 \le \delta \le d}, c'' \right)$. Not only the bases, also the subspaces $U_0' \subset V_0$ and $U_0'' \subset V_0$ may differ.

As in §4.3, we determine the matrix $B^{(\delta)}$ of the pairwise scalar products $\left\langle b_k'^{(\delta)}, \overline{b_\ell''^{(\delta)}} \right\rangle$ recursively by means of (4.2). For the start we need $B^{(0)}$, i.e., the scalar products $\int_0^{1/n} b_k'^{(0)}(x) \overline{b_\ell''^{(0)}}(x) \mathrm{d}x$ of the functions $b_k'^{(0)}, b_\ell''^{(0)} \in C([0,1/n])$. If, by some reason, the exact scalar product is not available, a numerical quadrature method may be used.

As soon as $B^{(d)}$ is computed, the product $\langle \mathbf{v}, \mathbf{w} \rangle$ is given by (4.1).

6.4 Convolution algorithm

When we discuss the convolution $\mathbf{u} := \mathbf{v} * \mathbf{w}$, we have to distinguish three hierarchical representations with the respective subspaces $U^{(j)}$, $U'^{(j)}$, $U''^{(j)}$ generated by the bases $\{b_1^{(j)}, \ldots, b_{r_j}^{(j)}\}$, $\{b_1'^{(j)}, \ldots, b_{r_j'}'^{(j)}\}$, $\{b_1''^{(j)}, \ldots, b_{r_j''}''^{(j)}\}$. We assume that the convolution of the basis functions $b_\nu'^{(0)} \in U_0'$ and $b_\mu''^{(0)} \in U_0''$ is explicitly known, i.e., there is an orthonormal basis of U_0 such that

$$b_\nu'^{(0)} \star b_\mu''^{(0)} = \sum_{m=1}^{2} \sum_{k=1}^{r_0} \beta_{ij,km}^{(\delta)} b_k^{(0)} \otimes b_m \tag{6.2}$$

with $b_k^{(0)} \in U_0'' \subset C([0,1/n])$. The latter representation is justified by Remark 6.2.

The convolution algorithm is now identical to the scheme of §5.

7 Generalisations

7.1 General hierarchical format

So far, we have discussed the special hierarchical format from §3.2.2. The general hierarchical format allows to choose a general binary dimension partition tree T (see §3.2.1).

For simplicity, we assume the situation $\mathbf{V} = \bigotimes_{j=1}^d \mathbb{K}^2$ with $n = 2^d$ from §2. The root of the tree T is by definition the set $\rho = \{1, \ldots, d\}$. Its sons $\alpha', \alpha'' \in T$ are vertices with $\rho = \alpha' \cup \alpha''$ (disjoint union). Hence, $\lambda' + \lambda'' = d$ for $\lambda' := \#\alpha'$ and $\lambda'' := \#\alpha''$. Set $n' := 2^{\lambda'}$, $n'' := 2^{\lambda''}$ and note that $n = n'n''$. Again, $\mathbb{K}^n$ is isomorphic to $\mathbb{K}^{n'} \otimes \mathbb{K}^{n''}$. If $n' > 2$, the vertex α' possesses two sons, i.e., $\mathbb{K}^{n'}$ is further split into $\mathbb{K}^{n_1} \otimes \mathbb{K}^{n_2}$ with $n' = n_1 n_2$; etc.

Each vertex $\alpha \in T$ is associated with the spaces $V_\alpha = \bigotimes_{j \in \alpha} \mathbb{K}^2$, $U_\alpha \subset U_{\alpha'} \otimes U_{\alpha''} \subset V_\alpha$ (see (3.2)). The bases of U_α, $U_{\alpha'}$, and $U_{\alpha''}$ are related by means of the coefficient matrices $C^{(\alpha,k)}$ (see (3.3)). Since, again, $U_\alpha = V_\alpha = \mathbb{K}^2$ holds for the leaves[9] $\alpha \in \mathcal{L}(T)$ (i.e., $\#\alpha = 1$), we fix the basis of U_α by (3.5). A tensor $\mathbf{v} \in \mathbf{V}$ represented in the hierarchical format is given by the data

$$\left(\left(C^{(\alpha,k)} \right)_{1 \le k \le r_\alpha, \alpha \in T \backslash \mathcal{L}(T)}, c \right),$$

i.e., $\mathbf{v} = \sum_{i=1}^{r_\rho} c_i b_i^{(\rho)}$ (ρ root of T), where the basis is defined recursively by (3.3) starting with $b_i^{(\alpha)} = b_i$ for $\alpha \in \mathcal{L}(T)$ (cf. (3.5)).

Next we discuss the *scalar product* of $\mathbf{v}, \mathbf{w} \in \mathbf{V}$ given by the data $\left(\left(C'^{(\alpha,k)} \right)_{1 \le k \le r'_\alpha, \alpha \in T \backslash \mathcal{L}(T)}, c' \right)$ and $\left(\left(C''^{(\alpha,k)} \right)_{1 \le k \le r''_\alpha, \alpha \in T \backslash \mathcal{L}(T)}, c'' \right)$. Again, we need the pairwise scalar products $\left\langle b_k'^{(\alpha)}, b_\ell''^{(\alpha)} \right\rangle$. The recursive computation uses

$$\left\langle b_k'^{(\alpha)}, b_\ell''^{(\alpha)} \right\rangle = \sum_{i=1}^{r'_{\alpha_1}} \sum_{j=1}^{r'_{\alpha_2}} \sum_{m=1}^{r''_{\alpha_1}} \sum_{n=1}^{r''_{\alpha_2}} c_{ij}'^{(\alpha,k)} \overline{c_{mn}''^{(\alpha,\ell)}} \left\langle b_i'^{(\alpha_1)}, b_m''^{(\alpha_1)} \right\rangle \left\langle b_j'^{(\alpha_2)}, b_n''^{(\alpha_2)} \right\rangle$$

[9] $\mathcal{L}(T)$ is the set of leaves of T, i.e., $\mathcal{L}(T) = \{\alpha \in T : \#\alpha = 1\}$.

(α_1 and α_2 sons of α) instead of (4.2). The final result is

$$\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{\ell=1}^{r'_\rho} \sum_{k=1}^{r''_\rho} c'^{(\rho)}_\ell \overline{c''^{(\rho)}_k} \left\langle b'^{(\rho)}_\ell, b''^{(\rho)}_k \right\rangle$$

(ρ root of T).

For the convolution, we consider the crucial isomorphism $\mathbb{K}^n \sim \mathbb{K}^{n'} \otimes \mathbb{K}^{n''} \subset \ell_0 \otimes \ell_0$ for $n = n'n''$ and the property that $v, w \in \mathbb{K}^n$ represented by elementary tensors $\mathbf{v} = v' \otimes v''$ and $\mathbf{w} = w' \otimes w''$ satisfy $\mathbf{v} \star \mathbf{w} \sim (v' \star w') \otimes (v'' \star w'')$. Vectors $v', w' \in \mathbb{K}^{n'}$ lead to $u := v' \star w' \in \mathbb{K}^{2n'-1} \subset \ell_0$. Writing $u \in \mathbb{K}^{2n'-1}$ as $u' + S^{n'}(u'')$ ($u', u'' \in \mathbb{K}^{n'}$, for the shift S see §2.2), we obtain

$$\mathbf{v} \star \mathbf{w} \sim u' \otimes (v'' \star w'') + u'' \otimes S^1 (v'' \star w'').$$

Since $v'' \star w'' \in \mathbb{K}^{2n'-1}$, it follows that $S^1 (v'' \star w'') \in \mathbb{K}^{2n'}$. Hence, there is a representation (5.8) for the convolution $b'^{(\alpha)}_i * b''^{(\alpha)}_j$ of the basis vectors from the respective subspaces U'_α and U''_α. The recursive computation of the coefficients $\beta^{(\alpha)}_{ij,km}$ in (5.8) is completely analogous to the procedure in §5.5.

7.2 Periodic convolution

The periodic convolution is $c_i = \sum_{k=0}^{n-1} a_k b_{i-k}$, where the indices are understood modulo n. Obviously, this kind of convolution cannot be performed direction-wise, since the direction-wise period would be 2. The simplest remedy is the periodisation of the previous convolution, i.e., perform $c' := a \star b \in \mathbb{K}^{2n-1}$ as before and set $c_i := c'_i + c'_{n+i}$ for $0 \le i \le n-1$, where $c_{2n-1} := 0$.

7.3 Matrix case

The isomorphism $\Phi_n : \bigotimes_{j=1}^d \mathbb{K}^2 \to \mathbb{K}^n$ can be easily extended to the matrix case: $\Phi_n : \bigotimes_{j=1}^d \mathbb{K}^{2\times 2} \to \mathbb{K}^{n \times n}$. Here, each matrix entry $M_{k\ell}$ with $k = \sum_{i=1}^d \kappa_i 2^{i-1}$ and $\ell = \sum_{j=1}^d \lambda_j 2^{j-1}$ corresponds to the tensor entry $\mathbf{M}_{(\kappa_1\lambda_1),\ldots,(\kappa_d\lambda_d)}$. Similarly, functions in two variables on $[0,1]^2$ can be understood as elements of a tensor space $\mathbf{V} = \bigotimes_{j=0}^d V_j$ with $V_0 = C([0,1/n]^2)$, $V_1 = \ldots = V_d = \mathbb{K}^{2\times 2}$. Now, Frobenius scalar products as well as convolutions in two variables can be performed.

The representation of a matrix $M \in \mathbb{K}^{n\times n}$ by a Kronecker-tensor $\mathbf{M} \in \bigotimes_{j=1}^d \mathbb{K}^{2\times 2}$ leads to the question how a matrix-vector multiplication $y := Mx$ is formed on the side of tensors. Let $\mathbf{x} \in \bigotimes_{j=1}^d \mathbb{K}^2$ and $x = \Phi_n \mathbf{x} \in \mathbb{K}^n$. Set $y := Mx$ and $\mathbf{y} := \Phi_n^{-1} y$. Since for fixed k we have

$$\mathbf{y}_{\kappa_1,\ldots,\kappa_d} = y_k = \sum_\ell M_{k\ell} x_\ell = \sum_{\lambda_1,\ldots,\lambda_d} \mathbf{M}_{(\kappa_1\lambda_1),\ldots,(\kappa_d\lambda_d)} \mathbf{x}_{\lambda_1,\ldots,\lambda_d},$$

where $k = \sum_{i=1}^d \kappa_i 2^{i-1}$ and $\ell = \sum_{j=1}^d \lambda_j 2^{j-1}$, we obtain $\mathbf{y} = \mathbf{M}\mathbf{x}$.

For elementary tensors $\mathbf{M} = \bigotimes_{j=1}^d M^{(j)}$ and $\mathbf{x} = \bigotimes_{j=1}^d x^{(j)}$, also $\mathbf{y} = \mathbf{M}\mathbf{x}$ is an elementary tensor $\bigotimes_{j=1}^d y^{(j)}$ with $y^{(j)} := M^{(j)} x^{(j)}$.

For r-term representations $\mathbf{M} = \sum_{\nu=1}^r \bigotimes_{j=1}^d M^{(j,\nu)}$ and $\mathbf{x} = \sum_{\mu=1}^s \bigotimes_{j=1}^d x^{(j,\mu)}$, we obtain an rs-term representation of the result $\mathbf{y} = \sum_{\nu=1}^r \sum_{\mu=1}^s \bigotimes_{j=1}^d M^{(j,\nu)} x^{(j,\mu)}$ involving drs matrix-vector multiplications $M^{(j,\nu)} x^{(j,\mu)}$, requiring $6drs$ arithmetical operations.

Finally, we discuss the use of the hierarchical format. Let $\mathbf{x}$ be represented by $\left\{ \left(C_x^{(\delta,k)} \right)_{1\le k\le r_\delta^x, 2\le\delta\le d}, c^x \right\}$ (cf. (3.7)). Analogously,[10] $\mathbf{M}$ is represented by $\left\{ \left(C_M^{(\delta,\ell)} \right)_{1\le \ell\le r_\delta^M, 2\le\delta\le d}, c^M \right\}$. The tensor $\mathbf{y}$ will be of the form $\left\{ \left(C_y^{(\delta,k)} \right)_{1\le k\le r_\delta^y, 2\le\delta\le d}, c^y \right\}$ with coefficients to be computed. In fact, the computation of $C_y^{(\delta,k)}$ follows the same lines as for the convolution in §5.5.

[10]Instead of (3.5) we use the fixed basis $\left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right\}$.

Let $\left\{ b_k^{x(\delta)} : 1 \leq k \leq r_\delta^x \right\} \subset \bigotimes_{j=1}^{\delta} \mathbb{K}^2$ be the basis of U_δ^x (cf. (3.4)), while $\left\{ b_\ell^{M(\delta)} : 1 \leq \ell \leq r_\delta^M \right\} \subset \bigotimes_{j=1}^{\delta} \mathbb{K}^{2\times 2}$ is the basis of U_δ^M. Then

$$U_\delta^y := \operatorname{span}\left\{ b_\ell^{M(\delta)} b_k^{x(\delta)} : 1 \leq k \leq r_\delta^x, 1 \leq \ell \leq r_\delta^M \right\} \subset \bigotimes_{j=1}^{\delta} \mathbb{K}^2$$

are the subspaces associated to $\mathbf{y}$. By induction, the products $b_\ell^{M(\delta-1)} b_k^{x(\delta-1)}$ can be represented in the basis $\left\{ b_i^{y(\delta-1)} : 1 \leq i \leq r_{\delta-1}^y \right\}$. Using the identities (3.6) for $b_\ell^{M(\delta)}$ and $b_k^{x(\delta)}$, the products $b_\ell^{M(\delta)} b_k^{x(\delta)}$ can be written as linear combinations of $b_i^{y(\delta-1)} \otimes b_j$. Orthonormalisation leads to a suitable basis $\left\{ b_i^{y(\delta)} \right\}$ of U_δ^y and to the coefficients in $b_\ell^{M(\delta)} b_k^{x(\delta)} = \sum_i \beta_{\ell k,i}^{(\delta)} b_i^{y(\delta)}$. The final result is $\mathbf{y} = \sum_i c_i^y b_i^{y(\delta)}$ with $c_i^y := \sum_{\ell,k} c_\ell^M \beta_{\ell k,i}^{(\delta)} c_k^x$.

References

[1] D. Braess and W. Hackbusch: *On the efficient computation of high-dimensional integrals and the approximation by exponential sums.* In: Multiscale, nonlinear and adaptive approximation (R. DeVore, A. Kunoth, eds.), Springer Berlin 2009, pp. 39-74.

[2] M. Espig: *Effiziente Bestapproximation mittels Summen von Elementartensoren in hohen Dimensionen.* Doctoral thesis, University Leipzig, 2008

[3] L. Grasedyck: *Polynomial approximation in hierarchical Tucker format by vector-tensorization.* Submitted 2010.

[4] W. Hackbusch: *Convolution of hp-functions on locally refined grids.* IMA J. Numer. Anal. **29** (2009) 960-985

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

[6] I.V. Oseledets and E.E. Tyrtyshnikov: *Breaking the curse of dimensionality, or how to use SVD in many dimensions.* SIAM J. Sci. Comput. **31** (2009) 3744-3759.