Numerical Tensor Techniques for Multidimensional Convolution Products

Wolfgang Hackbusch

Abstract

In order to treat high-dimensional problems, one has to find data-sparse representations. Starting with a six-dimensional problem, we first introduce the low-rank approximation of matrices. One purpose is the reduction of memory requirements, another advantage is that now vector operations instead of matrix operations can be applied. In the considered problem, the vectors correspond to grid functions defined on a three-dimensional grid. This leads to the next separation: these grid functions are tensors in $\mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$ and can be represented by the hierarchical tensor format. Typical operations as the Hadamard product and the convolution are now reduced to operations between $\mathbb{R}^n$ vectors. Standard algorithms for operations with vectors from $\mathbb{R}^n$ are of order $O(n)$ or larger. The tensorisation method is a representation method introducing additional data-sparsity. In many cases, the data size can be reduced from $O(n)$ to $O(\log n)$. Even more important, operations as the convolution can be performed with a cost corresponding to these data sizes.

Keywords  Tensorisation · Convolution · Tensor representation · Hierarchical representation

Mathematics Subject Classification (2010)  15A69 · 15A99 · 44A35 · 65F99 · 65T99

1 Introduction

In this paper, we recapitulate the numerical techniques which are needed to handle high-dimensional problems. As discussion starter, we use an example from quantum chemistry. The following function $h$ is to be determined:

$$h(x, z) = \int_{\mathbb{R}^3} f(x, x - y) g(y, z) \, dy \quad (x, z \in \mathbb{R}^3)$$

In memory of Eberhard Zeidler.

Wolfgang Hackbusch
wh@mis.mpg.de

1 Max-Planck-Institut Mathematik in den Naturwissenschaften, Inselstr. 22, D-04103, Leipzig, Germany
(for instance, \( f \) and \( g \) describe the pair amplitude and the pair interaction; cf. Flad–Flad-Harutyunyan [5]). A discretisation by a uniform grid \( \{ h = (i_1 h, i_2 h, i_3 h) : 0 \leq i_1, i_2, i_3 \leq n - 1 \} \) \((h: \text{grid size})\) in a cube leads to the discrete problem

\[
h_{ik} = h^3 \sum_j f_{i,j-k} g_{j,k} \quad (i = (i_1, i_2, i_3), \ k = (k_1, k_2, k_3), \ 0 \leq i_v, k_v \leq n - 1).
\]

(2)

Equation (2) describes an unusual matrix multiplication of convolution type:

\[
H = F \ast G \quad (H = (h_{ik}), \ F = (f_{i,j}), \ G = (g_{j,k})).
\]

(3)

The size of the matrices (number of entries) is \( n^6 \). Taking \( n \) of the size \( 2^{10} \approx 10^3 \) to \( 2^{20} \approx 10^6 \), it becomes obvious that naive methods cannot be used to perform the multiplication (3).

In Section 2, we shall consider the matrices in (3) as tensors of the space \( \mathbb{R}^N \otimes \mathbb{R}^N \) with \( N = n^3 \).

Then, problem (3) reduces to operations of vectors in \( \mathbb{R}^N \).

In a second step (Section 3), \( \mathbb{R}^N \) is regarded as the tensor space \( \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n \). For such tensors, we describe an efficient representation and show how operations are performed. In our example, we need two operations in \( \mathbb{R}^n \):

- the Hadamard product \( v \odot w \) defined by the componentwise product \( (v \odot w)_i = v_i w_i \), and
- the convolution \( v \ast w \) defined by \( (v \ast w)_i = \sum_\ell v_{i-\ell} w_\ell \).

The convolution \( v \ast w \) is a discretisation of the convolution of functions, \( \int_\mathbb{R} v(x - y)w(y)dy \), provided that \( v_i \) (\( w_i \)) are the nodal values of \( v \) (\( w \)) in an equidistant grid. For instance, the convolution in \( \mathbb{R}^n \) can be performed by the fast Fourier transform (FFT) requiring \( O(n \log n) \) operations. However, as explained in Section 4, we can perform the convolution (as well as the Hadamard product) much faster using the tensorisation technique. Here, \( \mathbb{R}^n \) for \( n = 2^L \) is replaced by the isomorphic tensor space \( \otimes^L \mathbb{R}^2 \). In many cases, grid functions in \( \mathbb{R}^n \)—in particular those from quantum chemistry—can be approximated by a tensor representation using only \( O(\log^* n) \) data. Then, the exact convolution of \( v \ast w \) requires not more than \( O(\log^* n) \) operations.

The convolution algorithm mentioned above is also interesting outside of quantum chemistry applications. Often, the functions \( v \) and \( w \) in \( \int_\mathbb{R} v(x - y)w(y)dy \) are represented by finite elements using locally refined grids or even \( hp \) techniques to reduce the number of degrees of freedom. If FFT is used for the convolution, one must transfer the finite-element functions to a uniform grid corresponding to the minimal grid size and thus one is destroying the advantages of the nonuniform finite-element approach. The tensorisation technique is able to represent the data at least as efficient as in the finite-element case. Then, the operation cost is determined by the data sizes of the representations. Moreover, it yields the optimal representation of the result \( v \ast w \).

---

1 Throughout the paper, \( \mathbb{R} \) may be replaced by \( \mathbb{C} \).
2 \( \log^*(n) \) denotes some (not specified) power of \( \log(n) \).
3 Appropriate algorithms are described in [7, 8].
2 Low-Rank Techniques for Matrices

2.1 Low-Rank Representation

In quantum chemistry, it is more usual to write the integral (1) as

\[ h(x, z) = \int_{\mathbb{R}^3} \tilde{f}(x, y)g(y, z)dy \quad (x, z \in \mathbb{R}^3) \]  

(4)

by introducing \( \tilde{f}(x, y) := f(x, x - y) \) (cf. [5, (1.4)]). Then, the discrete analogue is the standard matrix product \( \tilde{F}G \) instead of (3). However, this notation is less appropriate since the properties of the function \( f \) and of the matrix \( F \) are swept under the carpet.

The function \( f \) has a (representation) rank \( r \) if \( f(x, y) = \sum_{\nu=1}^{r} a_{\nu}(x)b_{\nu}(y) \), where \( \{a_{\nu}\} \) and \( \{b_{\nu}\} \) are linearly independent univariate functions. The latter identity is also written in tensor form as

\[ f = \sum_{\nu=1}^{r} a_{\nu} \otimes b_{\nu}. \]  

(5)

For instance, the function \( f(x, y) = \varphi(x)/\|y - y_0\| \) (\( y_0 \) position of a nucleus) has rank \( r = 1 \). However, the function \( \tilde{f}(x, y) := \varphi(x)/\|y_0 + x - y\| \) involved in (4) has infinite rank.

If the matrix \( F \in \mathbb{R}^{N \times N} \) has the rank \( r \), it allows a representation \( F = \sum_{\nu=1}^{r} a_{\nu}b_{\nu}^T \) \( (a_{\nu}, b_{\nu} \in \mathbb{R}^N) \). Again, we write

\[ F = \sum_{\nu=1}^{r} a_{\nu} \otimes b_{\nu}. \]  

(5)

The splitting of the tensor space \( \mathbb{R}^N \otimes \mathbb{R}^N \cong \mathbb{R}^{N \times N} \) (\( \cong \) denotes isomorphy) into the two factors \( \mathbb{R}^N \) is depicted in Fig. 1. In general, the tensor product \( v = v^{(1)} \otimes v^{(2)} \otimes \cdots \otimes v^{(d)} \) with \( v^{(j)} \in \mathbb{R}^{n_j} \) is a quantity indexed by \( d \)-tuples \( i = (i_1, \ldots, i_d) \) with the values

\[ v[i] = v^{(1)}[i_1] \cdot v^{(2)}[i_2] \cdot \cdots \cdot v^{(d)}[i_d] \quad (1 \leq i_j \leq n_j). \]  

(6)

Here and in the sequel, we use boldface letters for tensors and tensor spaces, while vectors, matrices, and vector spaces are denoted by standard letters.

If \( r \) is much smaller than \( N \), (5) describes the low-rank representation of \( F \). Note that the right-hand side of (5) requires only \( 2rN \ll N^2 \) data.

\( v^{(1)} \otimes v^{(2)} \otimes \cdots \otimes v^{(d)} \) is called an elementary tensor. In general, \( v^{(j)} \) may be elements of arbitrary vector spaces \( V_j \). The (algebraic) tensor space \( V = V_1 \otimes V_2 \otimes \cdots \otimes V_d = \bigotimes_{j=1}^{d} V_j \) is defined as the span of all elementary tensors (cf. [10, Section 3.2]).

Remark 1 As a consequence, linear maps on \( V \) are uniquely defined by their values of elementary tensors. The same holds for bilinear maps on Cartesian products \( V \times W \) of two tensor spaces.

Fig. 1 Tensor space \( \mathbb{R}^N \otimes \mathbb{R}^N \cong \mathbb{R}^{N \times N} \) and its factors \( \mathbb{R}^N, \mathbb{R}^N \).
2.2 SVD Truncation

Even if \( F \) has maximal rank \( N \), it might be well approximated by a low-rank matrix \( F_\varepsilon \) with rank \( r_\varepsilon \). For the precise analysis, we need the singular-value decomposition (SVD) of \( F \) which is

\[
F = \sum_{\nu=1}^{r} \sigma_{\nu} a_{\nu} \otimes b_{\nu}, \quad \{a_{\nu}\}, \{b_{\nu}\} \text{ orthonormal systems},
\]

with the singular values \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 \). The traditional formulation is \( F = U \Sigma V^T \), where the columns of \( U \) and \( V \) are defined by \( a_{\nu} \) and \( b_{\nu} \), respectively, and \( \Sigma \) is the diagonal matrix containing the singular values.

If \( \sigma_{r_\varepsilon} \leq \varepsilon \) for some \( r_\varepsilon < r \), the truncated matrix \( F_\varepsilon := \sum_{\nu=1}^{r_\varepsilon} \sigma_{\nu} a_{\nu} \otimes b_{\nu} \) has rank \( r_\varepsilon \) and satisfies the spectral norm estimate \( \|F - F_\varepsilon\|_2 \leq \varepsilon \).

Now, we assume \( F = \sum_{\nu=1}^{r} a_{\nu} \otimes b_{\nu} \) for the matrices in (3). We denote the entries of the vectors \( a_{\nu}, b_{\nu}, \ldots \) by \( a_{\nu}[i], b_{\nu}[i], \ldots \), where \( i \) abbreviates the triple \((i_1, i_2, i_3)\). Since \( F_{i,j} = \sum_{\nu=1}^{r} a_{\nu}[i]b_{\nu}[j] \) etc., the operation described in (2) becomes

\[
h_{ik} = h^3 \sum_{\nu=1}^{r} \sum_{\mu=1}^{s} a_{\nu}[i]b_{\nu}[i-j]c_{\mu}[j]d_{\mu}[k].
\]

\[
\sum_{j} b_{\nu}[i-j]c_{\mu}[j] \text{ is the component of the convolution } b_{\nu} \ast c_{\mu} \text{ at index } i. \text{ Set } q_{\nu\mu} := b_{\nu} \ast c_{\mu}. \text{ Then, the expression } \sum_{j} a_{\nu}[i]b_{\nu}[i-j]c_{\mu}[j] \text{ is the } i\text{-component of the Hadamard product } a_{\nu} \odot q_{\nu\mu}. \text{ Together, we obtain the representation of the matrix } H \text{ in (3) by}
\]

\[
H = \sum_{\mu=1}^{s} \left( h^3 \sum_{\nu=1}^{r} \left[ a_{\nu} \odot \left( b_{\nu} \ast c_{\mu} \right) \right] \right) \otimes d_{\mu}.
\]

Hence, the following has to be calculated:

(a) determine the vectors \( q_{\nu\mu} := b_{\nu} \ast c_{\mu} \in \mathbb{R}^N \),
(b) calculate the Hadamard products \( a_{\nu} \odot q_{\nu\mu} \in \mathbb{R}^N \),
(c) determine the sum \( e_{\mu} := h^3 \sum_{\nu=1}^{r} a_{\nu} \odot q_{\nu\mu} \).

Then, \( H = \sum_{\mu=1}^{s} e_{\mu} \otimes d_{\mu} \) is the representation of the resulting matrix. This shows that \( H \) is again a low-rank matrix if \( G \) is so. Nevertheless, one may apply a singular-value decomposition and truncate \( H \) to a lower rank.

Since \( N = n^3 \) holds with a large value of \( n \), even the simple Hadamard product in Step (b) is too costly when using the standard vector format. Instead we shall exploit the tensor structure of \( \mathbb{R}^N \).

For later use, we return to the representation (5). Let

\[
U := \text{span}\{a_{\nu} : 1 \leq \nu \leq r\}, \quad V := \text{span}\{b_{\nu} : 1 \leq \nu \leq r\}.
\]

Then, the tensor (matrix) \( F \) satisfies

\[
F \in U \otimes V \quad \text{with } \dim(U) = \dim(V) = r.
\]
Comparing (8) with \( F \in \mathbb{R}^N \otimes \mathbb{R}^N \), we see that the full space \( \mathbb{R}^N \) of dimension \( N \) is replaced by subspaces of dimension \( r \ll N \).

### 3 The Hierarchical Tensor Format

#### 3.1 Separation and Bilinear Operations

Here, we make use of the Cartesian product structure of the grid \( ((i_1h, i_2h, i_3h) : 0 \leq i_1, i_2, i_3 \leq n - 1) \). The tensor product of three vectors \( a, b, c \in \mathbb{R}^n \) is defined in (6). These tensors span the tensor space \( \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n \) which is isomorphic to \( \mathbb{R}^N \) (both spaces have dimension \( N = n^3 \)).

The analogue of the decomposition (5) would be the representation of \( v \in V := \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n \) by

\[
v = \sum_{v=1}^{r} a_v \otimes b_v \otimes c_v. \tag{9}
\]

The smallest possible value of \( r \) is called the rank of the tensor \( v \). The fact that in general the determination of this rank is NP hard (cf. Håstad [12]) already shows that the case of tensors of order \( \geq 3 \) is much more involved. In particular, there is no direct analogue of the singular-value decomposition. This leads to difficulties when one wants to truncate a tensor to lower order (cf. Espig–Hackbusch [4]).

The Hadamard product (componentwise product) \( \odot \) is a bilinear operation \( V \times V \rightarrow V \). Another bilinear map is the matrix-vector multiplication. For a unified approach let \( \boxtimes \) be the symbol of a general bilinear operation between two tensor spaces. An efficient computation of such a tensor operation \( \boxtimes : X \times Y \rightarrow Z \) (with \( X = \bigotimes_{j=1}^{d} X_j \), etc.) can be based on the following property (10), provided this property holds. Let \( x = \bigotimes_{j=1}^{d} x^{(j)} \) and \( y = \bigotimes_{j=1}^{d} y^{(j)} \) be elementary tensors\(^4\) with \( x^{(j)} \in X_j, y^{(j)} \in Y_j \). Then,

\[
\left( \bigotimes_{j=1}^{d} x^{(j)} \right) \boxtimes \left( \bigotimes_{j=1}^{d} y^{(j)} \right) = \bigotimes_{j=1}^{d} \left( x^{(j)} \boxtimes y^{(j)} \right)
\]

reduces the operation \( \boxtimes \) to simpler bilinear operations \( \boxtimes_j : X_j \times Y_j \rightarrow Z_j \) on the individual vector spaces.

In the case of the Hadamard product, \( \boxtimes = \odot \) is the componentwise product of tensors, while \( \boxtimes_j = \odot \) is the componentwise product of vectors. In fact, the property

\[
(a \otimes b \otimes c) \odot (a' \otimes b' \otimes c') = (a \odot a') \otimes (b \odot b') \otimes (c \odot c')
\]

follows since \( \{(a \otimes b \otimes c) \odot (a' \otimes b' \otimes c')\}_{[i]} = (a \otimes b \otimes c)_{[i]} \cdot (a' \otimes b' \otimes c')_{[i]} = a[i_1]b[i_2]c[i_3]a'[i_1]b'[i_2]c'[i_3] \) and \( \{(a \odot a') \otimes (b \odot b') \otimes (c \odot c')\}_{[i]} = (a \odot a')_{[i]}(b \odot b')(i_2)(c \odot c')(i_3) = a[i_1]a'[i_1]b[i_2]b'[i_2]c[i_3]c'[i_3] \) coincide. Note that on the left-hand side of (11) \( \odot \) acts on \( V \times V \), whereas on the right-hand side \( \odot \) acts on \( \mathbb{R}^n \times \mathbb{R}^n \).

Another example is the canonical scalar product of a (pre-)Hilbert tensor space \( X \) satisfying

\[
\left\langle \bigotimes_{j=1}^{d} x^{(j)}, \bigotimes_{j=1}^{d} y^{(j)} \right\rangle = \prod_{j=1}^{d} \langle x^{(j)} , y^{(j)} \rangle
\]

\(^4\)According to Remark 1, it is sufficient to investigate the mapping for elementary tensors.
This corresponds to (10) with \( Y = X \) and \( Z = \mathbb{R} \) (the field \( \mathbb{R} \) is considered as a tensor space of order \( d = 0 \)).

The notation \( (x \star y)[i] = \sum_j x[i - j]y[j] \) of the multivariate convolution involving multiindices \( i \in \mathbb{N}_0^d \) shows that also \( \Box = \ast \) satisfies (10). For \( d = 3 \), we have

\[
(a_v \otimes b_v \otimes c_v) \star (a_v' \otimes b_v' \otimes c_v') = (a_v \ast a_v') \otimes (b_v \ast b_v') \otimes (c_v \ast c_v').
\]

(12)

Hence, the Hadamard and convolution operations can be reduced to operations acting on vectors in \( \mathbb{R}^n \). If \( v \) and \( w \) are given in the form (9), all pairs of elementary terms can be treated by (11) or (12), respectively.

3.2 Introduction of the Hierarchical Format

In the following, we use the hierarchical format, which has the additional advantage that a SVD truncation can be performed (cf. [10, Section 11]). For that purpose, we need tensors of order 2 (matrix case) and rewrite \( \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n \) as \( (\mathbb{R}^n \otimes \mathbb{R}^n) \otimes \mathbb{R}^n \cong \mathbb{R}^{n^2} \otimes \mathbb{R}^n \). In a second step, we split \( \mathbb{R}^{n^2} \) into \( \mathbb{R}^n \otimes \mathbb{R}^n \). This leads to the binary tree shown in Fig. 2.

In the first step, we regard the components \( v[i] = v[i_1, i_2, i_3] \) of \( v \in \mathbb{R}^n \) as entries \( V[i_1, i_2, i_3] \) of the matrix \( V \in \mathbb{R}^{n^2 \times n} \cong \mathbb{R}^{n^2} \otimes \mathbb{R}^n \). As in Section 2, we may write \( V \) as \( \sum_{v=1}^n v^{(1)}_v \otimes v^{(2)}_v \) (cf. (5)) with \( v^{(1)}_v \otimes v^{(2)}_v \in \mathbb{R}^{n^2} \) and \( v^{(3)}_v \in \mathbb{R}^n \). In the second step, we regard \( v^{(1)}_v \) as \( n \times n \) matrices or equivalently as tensors of \( \mathbb{R}^n \otimes \mathbb{R}^n \) of the form \( \sum_{v=1}^n v^{(1)}_v \otimes v^{(2)}_v \).

Combining the structures of Figs. 1 and 2 yields the splitting depicted in Fig. 3. At the top of the tree, we see the matrix space \( \mathbb{R}^{N \times N} \cong \mathbb{R}^N \otimes \mathbb{R}^N \) with the sons \( \mathbb{R}^N \) on both sides. \( \mathbb{R}^N \cong \mathbb{R}^{n^2} \otimes \mathbb{R}^n \) is split into \( \mathbb{R}^{n^2} \) and \( \mathbb{R}^n \). Finally, \( \mathbb{R}^{n^2} \cong \mathbb{R}^n \otimes \mathbb{R}^n \) is split in two factors \( \mathbb{R}^n \).

Following the construction (8), we associate each vertex of the tree with a subspace. The leaves of the tree correspond to \( \mathbb{R}^n \). Therefore, there are six subspaces \( U_1, \ldots, U_6 \subset \mathbb{R}^n \). \( U_{12} \) and \( U_{45} \) are subspaces of \( \mathbb{R}^n \otimes \mathbb{R}^n \cong \mathbb{R}^{n^2} \), while \( U_{123} \) and \( U_{456} \) are subspaces of \( \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n \cong \mathbb{R}^N \). Also, the root \( \mathbb{R}^N \) has a subspace \( U_{1-6} \). The hierarchical structure is given by

\[
U_\alpha \subset U_{\alpha 1} \otimes U_{\alpha 2} \quad (\alpha_1, \alpha_2 \text{ sons of } \alpha),
\]

(13)

where \( \alpha \) belongs to the index set \( \{12, 123, 45, 456, 1-6\} \), i.e., \( U_{12} \subset U_1 \otimes U_2 \), \( U_{123} \subset U_{12} \otimes U_3 \), \ldots, \( U_{1-6} \subset U_{123} \otimes U_{456} \) (cf. Fig. 4). The condition (8) becomes

\[
F \in U_{1-6} \quad (1-6 \text{ is the index of the root}).
\]

(14)

The subspaces are (in principle) described by a basis (or at least a generating system). The bases of \( U_1, \ldots, U_6 \) corresponding to the leaves must be given explicitly. For the other
indices, we avoid an explicit description since the basis vectors of $\mathbb{R}^{n^2}$, $\mathbb{R}^{n^3}$, etc. are too large. Instead, we make use of (13). Let $\alpha$ be an index of an inner vertex of the tree (no leaf) and $\alpha_1$, $\alpha_2$ its sons. Let $\{b_{i}^{(\alpha_1)}: 1 \leq i \leq r_{\alpha_1}\}$ and $\{b_{j}^{(\alpha_2)}: 1 \leq j \leq r_{\alpha_2}\}$ be the bases of $U_{\alpha_1}$ and $U_{\alpha_2}$. Then $\{b_{i}^{(\alpha_1)} \otimes b_{j}^{(\alpha_2)}: 1 \leq i \leq r_{\alpha_1}, 1 \leq j \leq r_{\alpha_2}\}$ is a basis of $U_{\alpha_1} \otimes U_{\alpha_2}$. A basis vector $b_{\ell}^{(\alpha)} \in U_{\alpha} \subset U_{\alpha_1} \otimes U_{\alpha_2}$ must have a representation

$$b_{\ell}^{(\alpha)} = \sum_{i,j} c_{ij}^{(\alpha,\ell)} b_{i}^{(\alpha_1)} \otimes b_{j}^{(\alpha_2)} \quad (15)$$

with coefficients $c_{ij}^{(\alpha,\ell)}$ forming an $r_{\alpha_1} \times r_{\alpha_2}$ matrix

$$C^{(\alpha,\ell)} = (c_{ij}^{(\alpha,\ell)}). \quad (16)$$

It is sufficient to store $C^{(\alpha,\ell)}$ instead of $b_{\ell}^{(\alpha)}$. Note that the necessary memory is independent of the vector size $n$. 

$$\text{Fig. 3} \quad \text{Decomposition of } \mathbb{R}^{N \times N}$$
If (14) holds, the subspace \( U_{1-6} \) can be reduced to the one-dimensional space \( U_{\text{root}} = \text{span}(F) \). Let \( b_1^{(\text{root})} \) be the only basis vector. Then, only one additional factor \( c_1^{(\text{root})} \) is needed to characterise
\[
F = c_1^{(\text{root})} b_1^{(\text{root})}.
\]

Remark 2 (a) In the given example, we have to store the bases of \( U_1, \ldots, U_6 \) with the memory size \( \sum_{j=1}^{6} n_j r_j \). The matrices \( C^{(\alpha, \ell)} \) require the memory size \( r_1 r_2 + r_4 r_3 + r_4 r_5 + r_5 r_6 + 1 \cdot r_1 r_2 r_3 + r_4 r_5 r_6 \). \( c_1^{(\text{root})} \) is only one real number. If \( n_j \leq n \) and \( r_j \leq r \), the required memory size is bounded by \( 6nr + 4r^3 + r^2 + 1 \).

(b) In the general case of tensors of order \( d \) (instead of 6 as above), the bound is \( dnr + (d-1)r^3 + 1 \).

Below, we shall demonstrate that we can perform the required operations although we only have an indirect access to the bases.

### 3.3 Matricisation

The above construction gives rise to two questions: Do subspaces with the properties (13), (14) exist and what are their dimensions
\[
r_{\alpha} = \dim(U_{\alpha})
\]
in the best case? The answer is given by the matricisation which maps a tensor isomorphically into a matrix. We explain this isomorphism for the example \( \alpha = 45 \). The tensor \( F \in \bigotimes_{j=1}^{6} \mathbb{R}^{n_j} \) has six indices (we write \( F[i_1, \ldots, i_6] \) instead of \( F[i_1, i_2, i_3, j_1, j_2, j_3] = F[i, j] \)). The matrix \( M^{(45)} \) is of the size \( \mathbb{R}^{n_4 n_5} \) and has the entries
\[
M^{(45)}[(i_4, i_5), (i_1, i_2, i_3, i_6)] := F[i_1, i_2, i_3, i_4, i_5, i_6].
\]
The subspace
\[
U_{45} := \text{range}(M^{(45)}) \quad \text{with } r_{45} = \dim(U_{45}) = \text{rank}(M^{(45)})
\]
is the smallest subspace satisfying (13) and (14). For a more general description of the minimal subspaces see \([10, \text{Section 6}]\).

For \( v \in \bigotimes_{j=1}^{d} \mathbb{R}^{n_j} \) let \( \emptyset \neq \alpha \subsetneq \{1, \ldots, d\} \) be a subset with the complement \( \alpha^c := \{1, \ldots, d\} \setminus \alpha \). In general, the minimal subspace \( U_{\alpha}^{\text{min}}(v) := \text{range}(M^{(\alpha)}(v)) \) involves the matricisation
\[
M^{(\alpha)} = M^{(\alpha)}(v) \text{ which is defined by } M^{(\alpha)}[(i_j)_{j \in \alpha}, (i_j)_{j \in \alpha^c}] = v[i_1, \ldots, i_d].
\]
Note that the index sets need not be ordered, since we only use properties of \( M^{(\alpha)} \) which do not depend on the ordering. The (matrix) rank of \( M^{(\alpha)} \) is called the \( \alpha \)-rank of \( v \) (cf. Hitchcock \([13]\)):
\[
\text{rank}_{\alpha}(v) := \text{rank}(M^{(\alpha)}(v)).
\]

### 3.4 Hadamard Product and General Bilinear Operations

In the following, the Hadamard product \( \odot \) can be replaced by a general bilinear operation \( \square \) (cf. (10)).

In (7), we need the Hadamard product \( v \odot w \) of two tensors in \( \bigotimes_{j=1}^{3} \mathbb{R}^{n_j} \). We assume that both \( v \) and \( w \) are represented in the hierarchical format corresponding to the tree depicted...
in Fig. 2. \( v \) uses the bases \( \{ b_i^{(j)} : 1 \leq i \leq r_j \}, 1 \leq j \leq 3, \) at the leaves and the coefficients \( c_{ij}^{(\alpha,\ell)}, c_1^{(\text{root})} \), whereas \( w \) is represented by \( \{ b_i^{(j)} \}, c_{ij}^{(\alpha,\ell)}, c_1^{(\text{root})} \). Also, the ranks \( r_\alpha \) and \( r'_\alpha \) may be different.

We start at the leaves and determine the Hadamard product of the basis vectors explicitly:

\[
b_i^{(j)}(i,i') := b_i^{(j)} \odot b_i^{(j)} \quad (1 \leq j \leq 3, 1 \leq i \leq r_j, 1 \leq i' \leq r'_j).
\]

By induction, we assume that the products \( b_i^{(j)}(i,i') \) and \( b_j^{(j')}(j,j') \) are (directly or indirectly) determined. Then, (15) and (11) prove that

\[
b_{(\ell,m)}^{(\alpha)} := b_{(\ell)}^{(\alpha)} \odot b_{(m)}^{(\alpha)} = \left( \sum_{i,j} c_{ij}^{(\alpha,\ell)} b_i^{(\alpha)} \otimes b_j^{(\alpha)} \right) \odot \left( \sum_{i',j'} c_{ij}^{(\alpha,m)} b_i^{(\alpha)} \otimes b_j^{(\alpha)} \right)
= \sum_{i,j} \sum_{i',j'} c_{ij}^{(\alpha,\ell)} c_{ij}^{(\alpha,m)} \left( b_i^{(\alpha)} \odot b_j^{(\alpha)} \right) \odot \left( b_{i'}^{(\alpha)} \odot b_{j'}^{(\alpha)} \right)
= \sum_{(i',j') (j,j')} c_{ij}^{(\alpha,\ell)} c_{ij}^{(\alpha,m)} b_{(i,i')}^{(\alpha)} \otimes b_{(j,j')}^{(\alpha)}.
\]

(18)

The result \( x := v \odot w \) is represented by the generating system \( \{ b_i^{(j)}(i,i') \}, 1 \leq j \leq 3, \) at the leaves. Here, the pairs \((i, i')\) are the indices; thus, the index set has the size \( r_j r'_j \).

The equation (15) for the new vector contains the coefficients \( c_{ij}^{(\alpha,(\ell,m))} := c_{ij}^{(\alpha,m)} c_{ij}^{(\alpha,m)} \).

The coefficient \( c_1^{(\text{root})} \) is \( c_1^{(\text{root})} c_1^{(\text{root})} \), since \( v \odot w = \left( c_1^{(\text{root})} b_1^{(\text{root})} \right) \odot \left( c_1^{(\text{root})} b_1^{(\text{root})} \right) = c_1^{(\text{root})} c_1^{(\text{root})} b_1^{(\text{root})} \odot b_1^{(\text{root})} = c_1^{(\text{root})} c_1^{(\text{root})} b_{(1,1)}^{(\alpha)}.

We call \( \{ b_i^{(j)}(i,i') \} \) a generating system (or frame) since these vectors are not necessarily linearly independent. If not, the system \( \{ b_i^{(j)}(i,i') \} \) is larger than necessary and we can shorten the system. Even if \( \{ b_i^{(j)}(i,i') \} \) forms a basis, the question remains whether we can truncate the basis within a given tolerance. This will be the subject of Section 3.6.

Remark 3 The computation of all \( b_i^{(j)}(i,i') \) requires \( 3nr_j r'_j \) multiplications. If all coefficients \( c_{ij}^{(\alpha,(\ell,m))} \) are computed explicitly, we need \( r_\alpha r'_\alpha r_{\alpha_1} r'_{\alpha_2} r'_{\alpha_2} \) multiplications. The resulting cost is the product of the data sizes of \( v \) and \( w \).

In Section 4, the ranks \( r'_\alpha, r'_{\alpha_1}, r'_{\alpha_2} \) will be equal to 2.

3.5 Scalar Product, Orthonormalisation, Transformations

As mentioned above, the linear independence of the new frame \( \{ b_i^{(j)}(i,i') \} \) has to be checked. This can be done by the QR algorithm, provided we are able to determine scalar products \( \left< b_i^{(j)}(i,i'), b_j^{(j')}(m,m') \right> \) of the vectors determined in (18). We simplify the notation (index \( i \) instead of \( (\ell, m) \)) and consider the bases \( \{ b_i^{(\alpha)} \} \) at the vertex \( \alpha \) and their connection by (15). We proceed from the leaves to the root as in Section 3.4.
At the leaves, the bases are explicitly given so that the scalar products
\[ \sigma_{ij}^{(a)} := \langle b_i^{(a)}, b_j^{(a)} \rangle \]  
(19)
can be determined as usual. As soon as \( \sigma_{ij}^{(a_1)} \) and \( \sigma_{ij}^{(a_2)} \) are known for the sons of \( \alpha \), \( \sigma_{\ell m}^{(a)} \) can be determined by
\[
\sigma_{\ell m}^{(a)} = \sum_{i,j} c_{ij}^{(a,\ell)} b_i^{(a_1)} \otimes b_j^{(a_2)} = \sum_{i,j} c_{ij}^{(a,\ell)} \sigma_{ii}^{(a_1)} \sigma_{jj}^{(a_2)},
\]
(20)
since the Euclidean scalar product satisfies the rule \( \langle v \otimes w, x \otimes y \rangle = \langle v, x \rangle \langle w, y \rangle \). The induction (20) terminates at the vertex \( \alpha \), where the scalar products (19) are desired.

Of particular interest are orthonormal bases: \( \sigma_{ij}^{(a)} = \delta_{ij} \). Using (15), we obtain the following result.

**Remark 4** Let \( \alpha \) be a non-leaf vertex. The basis \( \{ b_i^{(a_1)} \} \) is orthonormal, if (a) the bases \( \{ b_i^{(a_1)} \} \) and \( \{ b_j^{(a_2)} \} \) of the sons \( \alpha_1, \alpha_2 \) are orthonormal and (b) the matrices \( C^{(\alpha,\ell)} \) in (16) are orthonormal with respect to the Frobenius scalar product: \( \langle C^{(\alpha,\ell)}, C^{(\alpha,m)} \rangle_F = \sum_{i,j} c_{ij}^{(\alpha,\ell)} c_{ij}^{(\alpha,m)} \delta_{\ell m} \).

The bases (or frames) can be orthonormalised as follows. Orthonormalise the explicitly given bases at the leaves (e.g., by QR). As soon as \( \{ b_i^{(a_1)} \} \) and \( \{ b_j^{(a_2)} \} \) are orthonormal, orthonormalise the matrices \( C^{(\alpha,\ell)} \). The new matrices \( C_{\text{new}}^{(\alpha,\ell)} \) define a new orthonormal basis \( \{ b_{i,\text{new}}^{(a)} \} \). The cost is described in [10, Remark 11.32].

The above mentioned calculations require basis transformations. Here, the following has to be taken into account (cf. [10, Section 11.3.1.4]).

- **Case A1.** Let \( \alpha_1 \) be the first son of \( \alpha \). Assume that the basis \( \{ b_i^{(a_1)} \} \) is transformed into a new basis \( \{ b_{i,\text{new}}^{(a_1)} \} \) so that \( b_i^{(a_1)} = \sum_k T_{ki} b_{k,\text{new}}^{(a_1)} \). Changing \( C^{(\alpha,\ell)} \) into \( C_{\text{new}}^{(\alpha,\ell)} := T C^{(\alpha,\ell)} T^T \), the basis \( \{ b_{\ell,\text{new}}^{(a)} \} \) remains unchanged.

- **Case A2.** If \( b_i^{(a_2)} = \sum_k T_{ki} b_{k,\text{new}}^{(a_2)} \) is a transformation of the second son of \( \alpha \), \( C^{(\alpha,\ell)} \) must be changed into \( C^{(\alpha,\ell)} T^T \).

- **Case B.** Consider a non-leaf vertex \( \alpha \). If the basis \( \{ b_{\ell}^{(a)} \} \) should be transformed into \( b_{\ell,\text{new}}^{(a)} := \sum_i T_{\ell i} b_i^{(a)} \), one has to change the coefficient matrices \( C^{(\alpha,\ell)} \) by \( C_{\text{new}}^{(\alpha,\ell)} := \sum_i T_{\ell i} C^{(\alpha,\ell)} \). (In addition, this transformation causes changes at the father vertex according to Case A1 or Case A2.)

### 3.6 SVD Truncation

The example in Section 3.4 shows that the Hadamard product is given by means of a generating system of increased size \( r_j r'_j \). This size may be larger than necessary and should be truncated. The truncation is prepared by an orthonormalisation as described in Section 3.5.
In principle, the SVD truncation is based on the singular-value decompositions of the matricisations\(^5\) \(M^{(\alpha)}\) (cf. Section 3.3). However, the singular values and singular vectors can be determined without the explicit knowledge of the huge matrix \(M^{(\alpha)}\).

Having generated orthonormal bases at all nodes, the singular-value decomposition starts at the root and proceeds to the leaves. It produces a basis \(\{b_{\ell,\text{new}}^{(\alpha)}\}\) together with singular values \(\sigma_{\ell}^{(\alpha)}\) indicating the importance of \(b_{\ell,\text{new}}^{(\alpha)}\). At the start \(\alpha = \text{root}\) there is only one (normalised) basis vector \(b_{\text{root}}^{(\alpha)} = b_{\text{root}}^{(\alpha)}\) which remains unchanged. The corresponding weight factor is \(\sigma_{\text{root}}^{(\alpha)} = |c_{\text{root}}^{(\alpha)}|\) (cf. (17)).

Assume that the new basis \(\{b_{\ell,\text{new}}^{(\alpha)}\}\) is already computed at the vertex \(\alpha\) and that \(\alpha\) is not a leaf but has sons \(\alpha_1, \alpha_2\). The basis \(\{b_{\ell,\text{new}}^{(\alpha)}\}\) is characterised by the matrices \(C^{(\alpha, \ell)}\). Together with the given values \(\sigma_{\ell}^{(\alpha)}\), we define the matrices\(^6\)

\[
Z_1 := \left[ \sigma_1^{(\alpha)} C^{(\alpha, 1)}, \sigma_2^{(\alpha)} C^{(\alpha, 2)}, \ldots, \sigma_{\alpha}^{(\alpha)} C^{(\alpha, r_\alpha)} \right] \in \mathbb{R}^{r_{\alpha_1} \times r_{\alpha_2}}, \\
Z_2 := \left[ \sigma_1^{(\alpha)} (C^{(\alpha, 1)})^T, \sigma_2^{(\alpha)} (C^{(\alpha, 2)})^T, \ldots, \sigma_{\alpha}^{(\alpha)} (C^{(\alpha, r_\alpha)})^T \right]^T \in \mathbb{R}^{(r_\alpha r_{\alpha_1}) \times r_{\alpha_2}}.
\]

The SVD of these matrices yields \(Z_1 = \sum_i \sigma_i^{(\alpha)} u_i^{(\alpha)} \otimes v_i^{(\alpha)}\) and \(Z_2 = \sum_i \sigma_i^{(\alpha)} u_i^{(\alpha)} \otimes v_i^{(\alpha)}\) with orthonormal vectors \(u_i^{(\alpha)} \in \mathbb{R}^{r_{\alpha_1}}\) and \(v_i^{(\alpha)} \in \mathbb{R}^{r_{\alpha_2}}\). Now, we have to transform the bases at the son nodes: \(\{b_{i,\text{new}}^{(\alpha_1)}\} := \{u_i^{(\alpha)}\}\) becomes the new basis for \(\alpha_1\), and \(\{b_{i,\text{new}}^{(\alpha_2)}\} := \{v_i^{(\alpha)}\}\) becomes the new basis for \(\alpha_2\). The new bases are called the HOSVD bases (cf. Footnote 5).

The procedure is repeated for the sons of \(\alpha_1, \alpha_2\) until we reach the leaves. Then, at all vertices, HOSVD bases are introduced together with singular values \(\sigma_i^{(\alpha)}\). As in Section 2.2, the SVD truncation consists of omitting all basis vectors corresponding to small enough singular values. Let \(\sigma_v^{(\alpha)}\), \(1 \leq v \leq r_\alpha\), be all singular values at \(\alpha\). Assume that we keep \(\sigma_v^{(\alpha)}\) for \(1 \leq v \leq s_\alpha\) and omit those for \(v > s_\alpha\). This means that (15) is reduced to \(b_i^{(\alpha)}\) with \(\ell \leq s_\alpha\) and that the double sum in (15) is taken over \(i \leq s_\alpha\) and \(j \leq s_\alpha\). Let \(v\) be the input tensor, while \(v_{\text{HOSVD}}\) denotes the truncated version. Then, the following estimate holds (cf. [10, Theorem 11.58]):

\[
\|v - v_{\text{HOSVD}}\| \leq \sqrt{\sum_{\alpha} \sum_{v \geq s_\alpha + 1} (\sigma_v^{(\alpha)})^2} \leq \sqrt{2d - 3}\|v - v_{\text{best}}\|.
\]

The first inequality allows us to explicitly control the error with respect to the Euclidean norm by the choice of the omitted singular values. The second inequality proves quasi-optimality of this truncation. \(v_{\text{best}}\) is the best approximation with the property that \(v_{\text{best}}\) satisfies \(\text{rank}_d(v_{\text{best}}) \leq s_\alpha\). The parameter \(d\) is the order of the tensor, i.e., \(d = 6\) in the case of Fig. 3 and \(d = 3\) for Fig. 2. Only in the (matrix) case of \(d = 2\), \(v_{\text{HOSVD}}\) coincides with \(v_{\text{best}}\).

\(^5\)Such SVDs are called the higher order singular-value decompositions (HOSVD) by De Lathauwer–De Moor–Vandevalle [2].

\(^6\)At the root, we have the special situation that \(Z_1 = Z_2\) because \(r_{\text{root}} = 1\).
3.7 Convolution

The treatment of Section 3.4 for the Hadamard operation \( \odot \) holds for any binary operation with the property (10). Because the multivariate convolution satisfies the analogous condition (12), the constructions of Section 3.4 also hold for the convolution \( \star \) instead of \( \odot \). Therefore, we can perform the convolution in \( \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n \cong \mathbb{R}^N \), provided that we are able to perform the convolution \((v \star w)_i = \sum_\ell v_{i-\ell} w_\ell \) in \( \mathbb{R}^n \).

The standard approach is the use of FFT (fast Fourier transform): First, the vectors \( v, w \) are mapped into their (discrete) Fourier images \( \hat{v}, \hat{w} \); then, the Hadamard product \( x := \hat{v} \odot \hat{w} \) is back-transformed into the convolution result \( \tilde{x} = v \star w \) (with suitable scaling). As well-known, the corresponding work is \( \mathcal{O}(n \log n) \). For large \( n \), this is still expensive. In the next chapter, we shall describe a much cheaper algorithm for \( v \star w \).

4 Tensorisation

The tensorisation has been introduced by Oseledets [17] (but for matrices instead of vectors). It is more natural to study this technique for vectors. The article Khoromskij [15]\(^7\) is the first one in this direction and contains several examples of this technique. Tensorisation\(^8\) together with truncation can be considered as an algebraic data compression method which is at least as successful as particular analytical compressions, e.g., by means of wavelets, hp methods. The analysis by Grasedyck [6] shows that under suitable conditions, the data size \( N(\tilde{v}_e) = \mathcal{O}(\log n) \) can be expected. Compression by tensorisation can be seen as a quite general multi-scale approach.

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 cost in the standard \( \mathbb{R}^n \) vector format. In particular, we discuss the Hadamard product and the (one-dimensional) 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 \( \tilde{v}_e \) and \( \tilde{w}_e \). The algorithm developed in Section 4.4 has a cost related to the data sizes \( N(\tilde{v}_e), N(\tilde{w}_e) \).

4.1 Grid Functions in \( \mathbb{R}^n \)

The following algorithms will apply to vectors in \( \mathbb{R}^n \) with \( n = 2^L \). The connection to the previous part is given by the fact that in Section 3 we have to perform various operations with the basis vectors \( b^{(j)}_i \in \mathbb{R}^n \). However, more general, the techniques of this chapter can be used for computations in \( \mathbb{R}^n \) without connection to the tensor problems in Sections 2 and 3.

Tensorisation is an interpretation of a usual \( \mathbb{R}^n \) vector as a tensor. Since \( n = 2^L \), there is a representation of the indices \( 0 \leq k \leq n - 1 \) by the binary numeral \((i_L, i_{L-1}, \ldots, i_1)_2\):

\[
k = \sum_{\ell=1}^L i_\ell 2^{\ell-1}, \quad i_\ell \in \{0, 1\}.
\]  

\(^7\)The preprint of [15] appeared in September 2009.

\(^8\)The term ‘tensorisation’ corresponds to the well-introduced terms ‘matricisation’ (mapping a tensor into a matrix, cf. Section 3.3) or ‘vectorisation’ (mapping a matrix or tensor into a vector).
We map the vector \( v \in \mathbb{R}^n \) into the tensor \( v \in \otimes^L \mathbb{R}^2 := \bigotimes_{j=1}^L \mathbb{R}^2 \) of order \( L \) by means of
\[
v[i_1, \ldots, i_L] = v_k \quad \text{with } k \text{ and } i_j \text{ as in (21)}. \tag{22}
\]
Since \( n = \dim(\mathbb{R}^n) = \dim(\otimes^L \mathbb{R}^2) = 2^L \), (22) describes an isomorphism
\[
\Phi : \otimes^L \mathbb{R}^2 \to \mathbb{R}^n, \quad v \mapsto v. \tag{23}
\]

On the side of tensors, we shall introduce a hierarchical tensor representation (cf. Section 3). This allows a simple truncation procedure \( v \mapsto v_\varepsilon \) (cf. Section 3.6). Often, the data size \( N(v_\varepsilon) \) of \( v_\varepsilon \) is much smaller than \( n \) (see Example 2). As a consequence, the tensorisation together with the truncation yields a black-box compression method for vectors in \( \mathbb{R}^n \).

### 4.2 TT Format

The underlying tree of the hierarchical representation is the linear tree\(^9\) depicted in Fig. 5. Hierarchical representations based on a linear tree are introduced by Oseledets [17] as TT format (cf. Oseledets–Tyrtyshnikov [18]). In principle, the hierarchical format requires subspaces at the leaves. Since \( \mathbb{R}^2 \) is extremely low-dimensional, we take the full space \( \mathbb{R}^2 \) and fix the basis by \( b^{(1)}_1 = (1,0) \) and \( b^{(1)}_2 = (0,1) \). Figure 5 corresponds to \( L = 4 \) (i.e., \( n = 16 \)). We replace the index \( \alpha = \{1,2,\ldots,\mu\} \) for the inner vertices by \( \mu \in \{2,\ldots,L\} \). The subspaces \( U_\mu \) belong to \( \bigotimes^\mu \mathbb{R}^2 \cong \mathbb{R}^{2^\mu} \) (in particular \( U_1 = \mathbb{R}^2 \)).

Since the TT-rank \( r_\mu = \text{rank}(M(\mu)) \) is the minimal dimension of the required subspace \( U_\mu \subset \bigotimes^\mu \mathbb{R}^2 \), the matricisation \( M(\mu) \) of a tensor \( v \) is of interest. In fact, \( M(\mu) \) can be expressed by means of the corresponding vector \( v = \Phi(v) \):

\[
M(\mu) = \begin{bmatrix}
v_0 & v_{2^\mu} & \ldots & v_{2^L-1} \\
v_1 & v_{2^\mu+1} & \ldots & v_{2^L-1+1} \\
\vdots & \vdots & \ddots & \vdots \\
v_{2^\mu-1} & v_{2^\mu+1-1} & \ldots & v_{2^{L-1}}
\end{bmatrix} \tag{24}
\]

\(^9\)All binary trees for tensors of order \( \leq 3 \) are linear trees, cf. Fig. 2.
Since we use the spaces $\mathbb{R}^2$ at the leaves, condition (13) becomes

$$U_{\mu+1} \subset U_\mu \otimes \mathbb{R}^2 \quad (1 \leq \mu \leq L - 1),$$

while (15) is

$$b^{(\mu+1)}_\ell = \sum_{i=1}^{r_\mu} \left[ c_{i1}^{(\mu+1,\ell)} b_i^{(\mu)} \otimes \binom{1}{0} + c_{i2}^{(\mu+1,\ell)} b_i^{(\mu)} \otimes \binom{0}{1} \right] \quad \text{for } 1 \leq \ell \leq r_{\mu+1}. \quad (26)$$

Before we discuss the operations, we want to show that grid functions appearing in practice may have ranks of the order $O(L) = O(\log n) \ll n$.

**Remark 5** Let $f$ be an analytic function in $(0, 1]$ with a singularity at $x = 0$. An efficient approximation is given by the $hp$ finite-element approach. In a simplified version, one uses polynomials of degree $g$ to interpolate $f$ in $[1/2, 1], [1/4, 1/2], \ldots, [2^{-L}, 2 \cdot 2^{-L}], [0, 2^{-L}]$. The data size is $D = (L+1)(g+1)$ since there are $L+1$ intervals and the polynomials have $g+1$ coefficients. For the typical asymptotically smooth functions (cf. [11, Appendix E]), one obtains an error estimate decaying exponentially in $D$. Let $F$ be the piecewise interpolation polynomial and evaluate $F$ at the equidistant grid points: $v_i := F(i \cdot 2^{-L})$ for $0 \leq i \leq n-1$. Inspection of the matrix $M^{(\mu)}$ shows that all columns except the first one contain grid values of a polynomial of degree $g$. Hence this part has at most the rank $g + 1$. The first column can increase the rank only by one so that $r_\mu = \text{rank}(M^{(\mu)}) \leq g + 2$. Therefore, the TT format representing $v = \Phi^{-1}(F)$ is of the same size as the $hp$ approach. The optimal approximation of $f$ by the TT format with rank$(M^{(\mu)}) \leq g + 2$ yields an error which is as most as large as the $hp$ error, i.e., it is exponentially decreasing with $g$. More details can be found in Grasedyck [6].

**Example 1** A particular function is the exponential $z^x$, where $z \neq 0$ may be any complex number. The grid values $v_i$ are $\zeta^i$ with $\zeta = z^{2^{-L}}$. For this vector, the columns of $M^{(\mu)}$ in (24) are linearly dependent so that rank$(M^{(\mu)}) = 1$. In fact, $v = \Phi^{-1}(v)$ is the elementary tensor $v = \bigotimes_{j=1}^L \binom{1}{\zeta^{2j-1}}$. Since $\sin(ax) = \frac{\exp(iax) - \exp(-iax)}{2i}$, any trigonometric function leads to rank$(M^{(\mu)}) = 2$.

This example (mentioned in [15]) implies the next remark.

**Remark 6** All functions with a limited number of exponential terms lead to a constant bound of rank$(M^{(\mu)})$ (e.g., $f(x) = \sum_{\nu=1}^r a_\nu \exp(-\beta_\nu x)$ yields rank$(M^{(\mu)}) \leq r$). A similar result holds for functions involving a fixed number of trigonometric terms (band-limited functions).

An example of a band-limited function can be found in Khoromskij–Veit [16].

The next example again shows that exponential sums can approximate functions with point singularities (Remark 5 is another approach to this problem). This fact is important for applications in quantum chemistry where singularities appear at the positions of the nuclei. This is an indication that the basis vectors appearing in $U_j$ $(1 \leq j \leq 6)$ for the problem (1) allow a tensorisation with moderate ranks.
Example 2 For \( n = 2^L \) set \( v = (f(k \cdot 2^{-L}))_{k=0}^{n-1} \in \mathbb{R}^n \) for the function \( f(x) = 1/(1 - x) \) in \([0, 1)\). For any \( r \in \mathbb{N} \), there is an approximation \( v_\mu \in \mathbb{R}^n \) such that \( v_\mu := \Phi^{-1}(v_\mu) \) yields ranks \( r_\mu = \text{rank}(M(\mu)) \leq r \) and satisfies the componentwise error estimate

\[
|v[k] - v_\mu[k]| \leq C_1 n \exp(-C_2 r) \quad \text{with } C_1, C_2 > 0 \text{ for all } 0 \leq k < n.
\]

Hence, for a given error bound \( \varepsilon > 0 \), the choice \( r = \mathcal{O}(\log(n) + \log \frac{1}{\varepsilon}) \) is sufficient. The storage size of the tensor \( v_\mu \) is \( \mathcal{O}(\log^2(n) + \log(n) \log \frac{1}{\varepsilon}) \).

Proof The function \( 1/t \) can be approximated in \([2^{-L}, 1]\) by an expression of the form \( \sum_{\nu=1}^{\nu} \alpha_\nu \exp(-\beta_\nu x) \). The error estimates follow from Braess–Hackbusch [1]. \( \square \)

4.3 Hadamard Product in \( \mathbb{R}^n \)

Since it does not matter whether the componentwise multiplication is realised via \( v_k \cdot w_k \) or \( v[i_1, \ldots, i_L] \cdot w[i_1, \ldots, i_L] \), the property (10) holds also in the case of the artificial tensor product \( \otimes L \mathbb{R}^2 \); more precisely,

\[
\Phi \left( \bigotimes_{j=1}^L v^{(j)} \right) \circ \Phi \left( \bigotimes_{j=1}^L w^{(j)} \right) = \Phi \left( \bigotimes_{j=1}^L (v^{(j)} \circ w^{(j)}) \right) = \Phi(v \circ w).
\]

Conclusion 1 Assume \( v = \Phi(v) \) and \( w = \Phi(w) \). Let \( v, w \) be represented by the TT format. Then the Hadamard product \( v \circ w \) can be computed as explained in Section 3.4. Since \( \Phi(v \circ w) = v \circ w \), the result is the tensorisation of \( v \circ w \). The computational cost is discussed in Section 3.4.

We return to the hierarchical format for true tensors as in Figs. 2 or 3. The subspaces at the leaves are described by bases containing \( \mathbb{R}^n \) vectors. The application of the tensorisation to these vectors corresponds to an extended tree as sketched in Fig. 6.

The combination of the tree in Fig. 2 with the TT tree corresponds to \( \mathbb{R}^N \cong \bigotimes^3 (\otimes L \mathbb{R}^2) \cong \bigotimes^3 L \mathbb{R}^2 \). For tensors represented in this format, we can again apply the algorithm in Section 3.4 to compute \( v \circ w \) for \( v, w \in \mathbb{R}^N \).

![Fig. 6 Extended tree](image-url)
4.4 Convolution in $\mathbb{R}^n$

4.4.1 Definition of the Convolution

We take a closer look to the convolution operation. The sum in $(v \star w)_i = \sum \ell v_{i-\ell} w_\ell$ is restricted to those $\ell$ with $0 \leq i - \ell, \ell \leq n - 1$, i.e.,

$$(v \star w)_i = \sum_{\ell = \max\{0,i+1-n\}}^{\min\{n-1,i\}} v_{i-\ell} w_\ell.$$

If $i$ varies in $[0, n - 1] \cap \mathbb{Z}$, the sum can be written as $\sum_{i=0}^i$. For $i < 0$, the empty sum yields $(v \star w)_i = 0$, but for $n \leq i \leq 2n - 2$, the sum in (27) is not empty. This shows the following remark.

Remark 7 The convolution of two $\mathbb{R}^n$ vectors yield an $\mathbb{R}^{2n-1}$ vector.

The notation becomes simpler if we replace the vector $v \in \mathbb{R}^n$ by the infinite sequence $v = (v_i)_{i \in \mathbb{N}_0}$ with $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ and $v_i := 0$ for all $i \geq n$. The set $\ell_0 = \ell_0(\mathbb{N}_0)$ consists of all sequences with only finitely many nonzero components. Now, the sum becomes

$$(v \star w)_i = \sum_{\ell=0}^i v_{i-\ell} w_\ell \quad \text{for all } i \in \mathbb{N}_0 \text{ and all } v, w \in \ell_0.$$

Remark 8 The $n$-periodic convolution is $(v \star_{\text{per}} w)_i = \sum_{\ell=0}^i v_{i-\ell} w_\ell (0 \leq i \leq n - 1)$, where all indices are understood modulo $n$. These values can be obtained by $(v \star_{\text{per}} w)_i = (v \star w)_i + (v \star w)_{n+i}$ for $0 \leq i \leq n - 1$.

4.4.2 Principal Idea of the Algorithm

For multivariate (grid) functions, the definition of the convolution implies the property (10): the convolution of elementary tensors can be reduced to the tensor product of one-dimensional convolutions.

Since now the vector $v$ is replaced by the tensor $v \in \bigotimes^L \mathbb{R}^2$, an obvious question is whether the product of $v = \bigotimes_{j=1}^L v^{(j)}$ and $w = \bigotimes_{j=1}^L w^{(j)}$ can be expressed by $x := \bigotimes_{j=1}^L (v^{(j)} \star w^{(j)})$ corresponding to (10), i.e., whether the corresponding vectors satisfy $\Phi(v) \star_{\text{per}} \Phi(w) = \Phi(x)$. In the naive sense, this cannot be true by the simple reason that $v^{(j)} \star w^{(j)}$ is a vector with three nontrivial components (cf. Remark 7). Therefore, the result does not belong to $\bigotimes^L \mathbb{R}^2$. Furthermore, we must expect a result in $\bigotimes^{L+1} \mathbb{R}^2$ since $v \star w$ has the length $2n - 1 > 2^L$ and $< 2^{L+1}$.

4.4.3 Extension to $\bigotimes^L \ell_0$

According to Section 4.4.1, $\mathbb{R}^2$ can be considered as a subspace of $\ell_0$. Hence, $\bigotimes^L \mathbb{R}^2$ is contained in $\bigotimes^L \ell_0$. The linear map $\Phi$ defined in (23) can be extended to $\Phi : \bigotimes^L \ell_0 \rightarrow \ell_0$ by

$$a = \Phi \left( \bigotimes_{j=1}^L v^{(j)} \right) \in \ell_0 \quad \text{with } a_k = \sum_{i_1, \ldots, i_L \in \mathbb{N}_0} \prod_{j=1}^d v^{(j)}[i_j]$$
Numerical Tensor Techniques for Multidimensional Convolution Products

(cf. Remark 1). In the case of \( v^{(j)} \in \mathbb{R}^2 \), the sum on the right-hand side of (28) contains only one term for \( 0 \leq k \leq n - 1 \) and the product \( \prod_{j=1}^{L} v^{(j)[i]} \) coincides with \( v[i_1, \ldots, i_L] \) for \( v := \bigotimes_{j=1}^{L} v^{(j)} \) (cf. (22)).

For a better understanding, we look at the case of \( L = 2 \).

Remark 9 Let \( e_i \in \ell_0 \) be the \( i \)th unit vector, i.e., \( e_i[j] = \delta_{ij} \) (\( i, j \in \mathbb{N}_0 \)). Then, \( b := \Phi(a \otimes e_i) \) is the vector \( a \in \ell_0 \) shifted by \( 2i \) positions: \( b_k := 0 \) for \( 0 \leq k < 2i \) and \( b_k = a_{k-2i} \) for \( k \geq i \).

The shift by \( p \) positions is denoted by \( S^p \). Thus, we can write \( b = S^{2i} a \).

### 4.4.4 Polynomials

Next, we use the isomorphism between \( \ell_0 \) and the space \( P \) of polynomials described by

\[ \pi : \ell_0 \to P \quad \text{with} \quad v \mapsto \pi[v](x) := \sum_{k \in \mathbb{N}_0} v_k x^k. \]

The connection with the convolution is given by the property that the product of two polynomials has the coefficients of the convolution product:

\[ \pi[v]\pi[w] = \pi[v \ast w] \quad \text{for} \quad v, w \in \ell_0. \] (29)

We define an extension of \( \pi : \ell_0 \to P \) to \( \hat{\pi} : \otimes^{L} \ell_0 \to P \) by

\[ \hat{\pi} : \otimes^{L} \ell_0 \to P \quad \text{with} \quad \hat{\pi} \left( \bigotimes_{j=1}^{L} v^{(j)} \right)(x) := \prod_{j=1}^{L} \pi[v^{(j)}](x^{2j-1}). \] (30)

A shift of \( v \) by \( i \) positions corresponds to the product \( \pi[S^i v] = \pi[v(x) \cdot x^i] \). This result together with Remark 9 shows that

\[ \hat{\pi} \left( \bigotimes_{j=1}^{L} v^{(j)} \right) = \pi \left( \Phi \left( \bigotimes_{j=1}^{L} v^{(j)} \right) \right). \] (31)

The extended map \( \Phi : \otimes^{L} \ell_0 \to \ell_0 \) is not injective. Two tensors \( v', v'' \in \otimes^{L} \ell_0 \) are called equivalent—denoted by \( v' \sim v'' \) — if they represent the same vector: \( \Phi(v') = \Phi(v'') \). From (31), we learn that the equivalence of \( v', v'' \) can also be expressed by \( \hat{\pi}[v'] = \hat{\pi}[v''] \).

By comparing the values under the map \( \hat{\pi} \), we obtain the following result.

**Lemma 1** \( \Phi \left( \bigotimes_{j=1}^{L} S_{m_j} v^{(j)} \right) = S^{m} \Phi \left( \bigotimes_{j=1}^{L} v^{(j)} \right) \) holds for \( m = \sum_{j=1}^{L} m_j 2^{j-1} \).

According to (10), we define the convolution of two (elementary) tensors in \( \otimes^{L} \ell_0 \) by

\[ \left( \bigotimes_{j=1}^{L} v^{(j)} \right) \ast \left( \bigotimes_{j=1}^{L} w^{(j)} \right) := \bigotimes_{j=1}^{L} \left( v^{(j)} \ast w^{(j)} \right). \] (32)

Now, the product \( v^{(j)} \ast w^{(j)} \) makes sense since it belongs to \( \ell_0 \). Next, we have to prove that the convolution introduced in (32) is consistent with the usual convolution of vectors.
Lemma 2 Let \( v = \Phi \left( \bigotimes_{j=1}^{L} v^{(j)} \right) \) and \( w = \Phi \left( \bigotimes_{j=1}^{L} w^{(j)} \right) \) be vectors in \( \ell_0 \). Then, (32) implies
\[
\Phi \left( \bigotimes_{j=1}^{L} (v^{(j)} \star w^{(j)}) \right) = v \star w.
\]

Proof. Since \( \pi : \ell_0 \to \mathbb{P} \) is an isomorphism, the statement is equivalent to \( \pi \left[ \Phi \left( \bigotimes_{j=1}^{L} (v^{(j)} \star w^{(j)}) \right) \right] = \pi [v \star w] \). The left-hand side of this equation is
\[
\pi \left[ \Phi \left( \bigotimes_{j=1}^{L} (v^{(j)} \star w^{(j)}) \right) \right] (x) \equiv \hat{\pi} \left[ \bigotimes_{j=1}^{L} (v^{(j)} \star w^{(j)}) \right] (x)
\]
\[
\equiv (30) \prod_{j=1}^{L} \pi [v^{(j)} \star w^{(j)}] (x^{2j-1})
\]
\[
\equiv (29) \prod_{j=1}^{L} \pi [v^{(j)}] (x^{2j-1}) \cdot \pi [w^{(j)}] (x^{2j-1})
\]
\[
= \left( \prod_{j=1}^{L} \pi [v^{(j)}] (x^{2j-1}) \right) \cdot \left( \prod_{j=1}^{L} \pi [w^{(j)}] (x^{2j-1}) \right)
\]
\[
= \hat{\pi} \left[ \bigotimes_{j=1}^{L} v^{(j)} \right] (x) \cdot \hat{\pi} \left[ \bigotimes_{j=1}^{L} w^{(j)} \right] (x)
\]
\[
\equiv (31) \pi [v] (x) \cdot \pi [w] (x) \equiv (29) \pi [v \star w] (x).
\]

\[ \square \]

4.5 Carry-over Procedure

The result \( \bigotimes_{j=1}^{L} (v^{(j)} \star w^{(j)}) \) is still unsatisfactory because \( v^{(j)}, w^{(j)} \in \mathbb{R}^2 \) produce \( v^{(j)} \star w^{(j)} \in \mathbb{R}^3 \). A solution can be as follows. Let \( L = 2 \) as in Remark 9. Consider \( a \otimes b \) with \( a, b \in \ell_0 \). We want to find an equivalent tensor with factors in \( \mathbb{R}^2 \). Assume that \( a_K \neq 0 \), but \( a_i = 0 \) for \( i > K \), which implies \( a \in \mathbb{R}^{K+1} \). If \( K = 1 \), \( a \) belongs to \( \mathbb{R}^2 \) and nothing has to be done. If \( K > 1 \) set \( a' \in \mathbb{R}^2 \) with \( a'_i = a_i \) for \( i = 0, 1 \) and \( a'' \in \ell_0 \) with \( a''_i = a_{i+2} \) for \( i \in \mathbb{N}_0 \), Using Remark 9, one checks that \( a \otimes b \) represents the same vector as \( a' \otimes b + a'' \otimes Sb \), where \( Sb \) is the shifted version of \( b \):
\[
\Phi (a \otimes b) = \Phi (a' \otimes b + a'' \otimes Sb).
\]

\( a' \in \mathbb{R}^2 \) is already of the desired form. \( a'' \) belongs to \( \mathbb{R}^{K-1} \). This procedure can again be applied to \( a'' \otimes b'' \) until all first factors belong to \( \mathbb{R}^2 \).

In the case of a general tensor \( \bigotimes_{j=1}^{L} v^{(j)} \), this procedure is applied to the first factor \( v^{(1)} \) and yields sums of elementary tensors of the form \( w^{(1)} \otimes \bigotimes_{j=2}^{L} w^{(j)} \) with \( w^{(1)} \in \mathbb{R}^2 \). Then, the procedure is repeated with the second factor resulting in sums of the terms \( x^{(1)} \otimes x^{(2)} \otimes \bigotimes_{j=3}^{L} x^{(j)} \) with \( x^{(1)}, x^{(2)} \in \mathbb{R}^2 \), etc. In the case of the last factor, we may
have to add an \((L+1)\)-th factor. Since we know that \(v \ast w\) belongs to \(\mathbb{R}^{2n-1}\), the \((L+1)\)-th factor must belong to \(\mathbb{R}^2\).

### 4.6 Convolution Algorithm

We recall Remark 7: If \(v, w \in \bigotimes_{j=1}^{L} \mathbb{R}^2\), the result is a tensor \(u := v \ast w\) in \(\bigotimes_{j=1}^{L+1} \mathbb{R}^2\). Lemma 3 describes the start at \(\delta = 1\), while Lemma 4 characterises the recursion. In the following, the vector notation \(v = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}\) means \(v_0 = \alpha, v_1 = \beta\), i.e., the components must be read from the top to the bottom. By \(v \sim w\), we denote the equivalence \(\Phi_1(v) = \Phi_1(w)\).

**Lemma 3** The convolution of \(v = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}\) and \(w = \begin{bmatrix} \gamma \\ \delta \end{bmatrix}\) yields

\[
\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \ast \begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} \alpha \gamma \\ \alpha \delta + \beta \gamma \\ \beta \delta \\ 0 \end{bmatrix} \sim \begin{bmatrix} \alpha \gamma \\ \alpha \delta + \beta \gamma \\ \beta \delta \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} \beta \delta \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \bigotimes_{j=1}^{2} \mathbb{R}^2. \tag{33a}
\]

Furthermore, the shifted vector has the tensor representation

\[
S \begin{bmatrix} \alpha \gamma \\ \alpha \delta + \beta \gamma \\ \beta \delta \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \alpha \gamma \\ \alpha \delta + \beta \gamma \\ \beta \delta \end{bmatrix} \sim \begin{bmatrix} 0 \\ \alpha \gamma \\ \alpha \delta + \beta \gamma \\ \beta \delta \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} \beta \delta \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \bigotimes_{j=1}^{2} \mathbb{R}^2. \tag{33b}
\]

The basic identity is given in the next lemma.

**Lemma 4** For given \(v, w \in \bigotimes_{j=1}^{\delta-1} \mathbb{R}^2\) let the convolution result be

\(v \ast w \sim a = a' \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + a'' \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \bigotimes_{j=1}^{\delta} \mathbb{R}^2. \tag{34a}\)

Then, convolution of the tensors \(v \otimes x\) and \(w \otimes y\) with \(x = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}\), \(y = \begin{bmatrix} \gamma \\ \delta \end{bmatrix}\) yields

\[
(v \otimes x) \ast (w \otimes y) \sim u = u' \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + u'' \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \bigotimes_{j=1}^{\delta+1} \mathbb{R}^2
\]

with

\[
u' = a' \otimes \begin{bmatrix} \alpha \gamma \\ \alpha \delta + \beta \gamma \\ \beta \delta \\ 0 \end{bmatrix} + a'' \otimes \begin{bmatrix} 0 \\ \alpha \gamma \\ \alpha \delta + \beta \gamma \\ \beta \delta \end{bmatrix} \in \bigotimes_{j=1}^{\delta} \mathbb{R}^2
\] \hspace{1cm} (34b)

and

\[
u'' = a' \otimes \begin{bmatrix} \beta \delta \\ 0 \end{bmatrix} + a'' \otimes \begin{bmatrix} \alpha \gamma \\ \alpha \delta + \beta \gamma \end{bmatrix} \in \bigotimes_{j=1}^{\delta} \mathbb{R}^2.
\]

**Proof** Lemma 2 implies that

\[
(v \otimes x) \ast (w \otimes y) \sim (v \ast w) \otimes z \quad \text{with} \quad z := x \ast y \in \mathbb{R}^3 \subset \ell_0.
\]

Assumption (34a) yields

\[
(v \ast w) \otimes z \sim \left( a' + S^{2^{\delta}} a'' \right) \otimes z.
\]
Lemma 1 shows that
\[ S^{2^{k-1}} a'' \otimes z = S^{2^{k-1}}(a'' \otimes z) \sim a'' \otimes (S_z). \]

Using (33a) and (33b), we obtain
\[
(a \otimes z) \sim a' \otimes \left[ a_{\delta + \beta y} \right] \otimes \left[ i \right] + a' \otimes \left[ i \right] + a'' \otimes \left[ \frac{0}{1} \right] \otimes \left[ i \right] , \\
(S^{2^{k-1}} a'') \otimes z \sim a'' \otimes (S_z) \sim a'' \otimes \left[ 0 \right] \otimes \left[ i \right] + a'' \otimes \left[ \frac{0}{1} \right] \otimes \left[ i \right] .
\]

Summation of both identities yields the assertion of the lemma.

If the vectors \( x, y \) in Lemma 4 belong to \( \left\{ \left[ 0 \right], \left[ i \right] \right\} \), the vectors \( \left[ a_{\delta + \beta y} \right], \left[ 0 \right], \left[ i \right] \), \( \left[ \frac{0}{1} \right] \) from (34b) belong to \( \{0, 1\} \).

Lemma 3 proves assumption (34a) for \( \delta = 2 \), while Lemma 4 shows that \( v \otimes x \) and \( w \otimes y \) satisfy the requirement (34a) for \( \delta + 1 \) instead of \( \delta \).

4.7 Convolution of Tensors in Hierarchical Format

We recall that the subspaces \( U_\delta \subset \otimes^\delta \mathbb{R}^2 \) satisfy (25): \( U_{\delta+1} \subset U_\delta \otimes \mathbb{R}^2 \). The essential observation is that also the results of the convolution yield subspaces with this property.

Note that there are three different tensors \( v, w, u := v \ast w \) involving representations with three different subspace families \( U'_{\delta}, U''_{\delta}, U_\delta \) \((1 \leq \delta \leq L)\). The bases spanning these subspaces consist of the vectors \( b_i^{\prime\prime}(\delta), b_i^{\prime}(\delta), b_i^{\prime\prime}(\delta) \). The dimensions of the subspaces are \( r'_\delta, r''_\delta, r_\delta \).

Any tensor \( a \in \otimes^\delta \mathbb{R}^2 \) can be written as \( a = a' \otimes \left[ 0 \right] + a'' \otimes \left[ i \right] \). Define the linear maps \( \phi'_\delta, \phi''_\delta : \otimes^\delta \mathbb{R}^2 \rightarrow \otimes^{\delta-1} \mathbb{R}^2 \) by \( \phi'_\delta(a) = a', \phi''_\delta(a) = a'' \).

**Theorem 2** Let the tensors \( v, w \in \bigotimes_{i=1}^L \mathbb{R}^2 \) be represented by (possibly different) hierarchical formats using the respective subspaces \( U'_\delta \) and \( U''_\delta \), \( 1 \leq \delta \leq L \), satisfying
\[
U'_1 = \mathbb{R}^2, \quad U''_1 \subset U'_{\delta-1} \otimes \mathbb{R}^2, \quad v \in U'_L, \\
U''_1 = \mathbb{R}^2, \quad U''_1 \subset U''_{\delta-1} \otimes \mathbb{R}^2, \quad w \in U''_L.
\]

The subspaces
\[
U_\delta := \text{span}(\phi'_{\delta+1}(x \ast y), \phi''_{\delta+1}(x \ast y) : x \in U'_\delta, y \in U''_\delta) \quad (1 \leq \delta \leq L)
\]
satisfy
\[
U_1 = \mathbb{R}^2, \quad U_\delta \subset U_{\delta-1} \otimes \mathbb{R}^2, \quad v \ast w \in U_{L+1}.
\]

The dimension of \( U_\delta \) can be bounded by
\[
\dim(U_\delta) \leq \min \left\{ 2 \dim(U'_\delta) \dim(U''_\delta), 2^\delta, 2^{L+1-\delta} \right\}.
\]

**Proof**

(i) \( U_1 = \mathbb{R}^2 \) can be concluded from Lemma 3.

(ii) Write \( x, y \in U'_\delta \subset U'_{\delta-1} \otimes \mathbb{R}^2 \) as \( x = x' \otimes \left[ 0 \right] + x'' \otimes \left[ i \right] \) and \( y = y' \otimes \left[ 0 \right] + y'' \otimes \left[ i \right] \), with \( x', x'', y', y'' \in U_{\delta-1} \). Expansion of the sums yields \( x \ast y = \left( x' \otimes \left[ 0 \right] \right) \ast \left( y' \otimes \left[ 0 \right] \right) + \cdots \). For each term \( z \) of this expansion, Lemma 4 (with \( v, w \) renamed \( x', x'' \)) states that \( \phi'_{\delta+1}(z) = u' \) and \( \phi''_{\delta+1}(z) = u'' \) belong to \( U_{\delta-1} \otimes \mathbb{R}^2 \) (cf. (34b)).
Hence, \( \phi^{\prime}_{\delta+1}(x \otimes y), \phi^{\prime\prime}_{\delta+1}(x \otimes y) \in U_{\delta-1} \otimes \mathbb{R}^2 \) holds, and the definition of \( U_\delta \) implies the inclusion \( U_\delta \subset U_{\delta-1} \otimes \mathbb{R}^2 \).

(iii) \( v \in U_L^\prime \) and \( w \in U_L^{\prime\prime} \) together with the definition of \( U_L \) lead to \( v \ast w \in U_L \).

(iv) The first bound of \( \dim(U_\delta) \) follows directly from (35b). The bound \( \min\{2^\delta, 2^{\delta+1-\delta} \} \) holds for any rank \( (M_1, \ldots, M_\delta)(v) \) of \( v \in \otimes \mathbb{R}^2 \).

The bound \( 2 \dim(U_\delta') \dim(U_\delta^{\prime\prime}) \) corresponds to the product mentioned in Remark 3.

For \( \delta = 1, \ldots, L \), the numerical scheme has

1. to introduce an orthonormal basis \( \{b_1^{(\delta)}, \ldots, b_r^{(\delta)}\} \) of \( U_\delta \), where \( r_\delta := \dim(U_\delta) \) (cf. Section 3.5),

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

\[
 b_i^{(\delta)} \ast b_j^{(\delta)} = \sum_{k=1}^{r_\delta} \sum_{m=1}^{2} \beta_{ij,km}^{(\delta)} b_k^{(\delta)} \otimes b_m. \tag{36}
\]

As soon as the \( \beta \)-coefficients from (36) are known, general products \( x \ast y \) of \( x \in U_\delta \) and \( y \in U_\delta^{\prime\prime} \) can be evaluated easily as shown in the next remark.

Remark 10 Let \( x = \sum_{i=1}^{r_\delta} \xi_i b_i^{(\delta)} \in U_\delta \) and \( y = \sum_{j=1}^{r_\delta} \eta_j b_j^{(\delta)} \in U_\delta^{\prime\prime} \). Then, convolution yields

\[
x \ast y = z = z' \otimes \begin{bmatrix} 1 \\ \xi_0 \end{bmatrix} + z'' \otimes \begin{bmatrix} 0 \\ \eta_0 \end{bmatrix}
\]

with \( z' = \sum_{k=1}^{r_\delta} \zeta_k (b_k^{(\delta)}), \ z'' = \sum_{k=1}^{r_\delta} \zeta_k'' (b_k^{(\delta)}), \)

where

\[
 \zeta_k' = \sum_{i=1}^{r_\delta} \sum_{j=1}^{r_\delta} \xi_i \eta_j \beta^{(\delta)}_{ij,k1}
\]

and

\[
 \zeta_k'' = \sum_{i=1}^{r_\delta} \sum_{j=1}^{r_\delta} \xi_i \eta_j \beta^{(\delta)}_{ij,k2}
\]

with \( \beta^{(\delta)}_{ij,km} \) from (36). The computation of \( \zeta_k', \zeta_k'' \) \((1 \leq k \leq r_\delta)\) requires \( 4r_\delta r_\delta' (r_\delta'' + 1) \) operations.

The total cost is described in [9, p. 482]. It is the sum

\[
 8r_\delta r_\delta' r_\delta'' (r_\delta'' + 1) + 8 (r_\delta' r_\delta'')^2 + 4 \frac{2}{3} (r_\delta' r_\delta'')^3 + 2r_\delta' r_\delta''^2 \quad \text{for } 2 \leq \delta \leq L. \tag{37}
\]

A rough estimate by \( r_\delta', r_\delta'' \leq r \) and \( r_\delta \leq 2r^2 \) yields the asymptotic bound \( \frac{100}{3} (L-1)r^6 \).

The higher order terms are caused by the orthonormalisation.

5 Toeplitz Matrices

5.1 Notation

A matrix \((a_{ij})\) is called a Toeplitz matrix if \(a_{ij}\) only depends on \(i-j\). A multiplication by a Toeplitz matrix and a convolution are almost equivalent (cf. Kazeev et al. [14]).

If we fix the vector \(x\) in \(x \otimes y\), this expression defines a linear map \(y \mapsto x \otimes y\) which may be expressed by a matrix \(T = T_x\), i.e., \(Ty := x \otimes y\). In the case of \(x, y \in \mathbb{R}^n\) and \(x \otimes y \in \mathbb{R}^{2n-1}\), \(T\) is the (rectangular) Toeplitz matrix of size \((2n-1) \times n\) with \(T_{ij} = x_i\) \((0 \leq i \leq n-1)\), \(T_{n-1+i,0} = T_0i = 0\) \(1 \leq i \leq n-1\).
A general \( n \times n \) Toeplitz matrix is uniquely determined by the coefficient vector \( a = [a_0, \ldots, a_{2n-2}] \):

\[
T(a) := \begin{bmatrix}
a_{n-1} & a_{n-2} & \cdots & a_0 \\
a_n & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & a_{n-2} \\
a_{2n-2} & \cdots & a_n & a_{n-1}
\end{bmatrix}, \quad \text{i.e., } T(a)_{i,j} = a_{n-1+i-j} \quad \text{for } 0 \leq i, j \leq n - 1.
\]

The product \( z := \alpha \ast \gamma \) belongs to \( \mathbb{R}^{3n-1} \). The part \( \tilde{z} \) with \( \tilde{z}_i := z_{n-1+i} \) \((0 \leq i \leq n - 1)\) coincides with \( T(a)y \in \mathbb{R}^n \).

### 5.2 Tensorisation for Matrices

The matrix space \( \mathbb{R}^{n \times n} \) for \( n = 2^L \) is isomorphic to \( \bigotimes_{j=1}^{L} \mathbb{R}^{2 \times 2} \). As in (23), the isomorphism \( \mathbf{M} \in \bigotimes_{j=1}^{L} \mathbb{R}^{2 \times 2} \mapsto \mathbf{M} \in \mathbb{R}^{n \times n} \) is defined by \( \mathbf{M}[i, j] = \mathbf{M}[(i_1, j_1), \ldots, (i_L, j_L)] \) where \( i = \sum_{\ell=1}^{L} i_\ell 2^{L-\ell}, \ j = \sum_{\ell=1}^{L} j_\ell 2^{L-\ell}, \ i_\ell, j_\ell \in \{0, 1\} \) (cf. [17]). In particular, a block matrix

\[
\begin{bmatrix}
M_{11} & M_{12} \\
M_{21} & M_{22}
\end{bmatrix}
\]

corresponds to the tensor product \( M_{11} \otimes \begin{bmatrix} 0 & 0 \\
0 & 0 \end{bmatrix} + M_{12} \otimes \begin{bmatrix} 0 & 1 \\
0 & 0 \end{bmatrix} + M_{21} \otimes \begin{bmatrix} 0 & 0 \\
1 & 0 \end{bmatrix} + M_{22} \otimes \begin{bmatrix} 0 & 0 \\
0 & 1 \end{bmatrix} \).

In the case of a Toeplitz matrix, all submatrices are again Toeplitz. In the previous example, \( M_{11} = M_{22} \) follows. Therefore, a suitable subspace \( U \) of \( \mathbb{R}^{2 \times 2} \) is spanned by

\[
\begin{bmatrix}
0 & 1 \\
0 & 0
\end{bmatrix}, \quad \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}, \quad \begin{bmatrix}
0 & 0 \\
1 & 0
\end{bmatrix}.
\]

For the hierarchical representation, we use the linear tree of Fig. 5 with \( \mathbb{R}^2 \) replaced by \( U \).

The TT-rank \( r_\mu = \dim(U_\mu) \) is described next. Let \( T = T(a) \in \mathbb{R}^{n \times n} \) be a Toeplitz matrix defined by the coefficient vector \( a \in \mathbb{R}^{2n-1} \) (cf. (38)). Consider a regular block structure of \( T \) with blocks of size \( 2^\mu \times 2^\mu \). Denote these blocks by \( T^{a_{\mu\beta}} = (T_{ij})_{2^\mu \leq i, j \leq 2^\mu+1} \) for \( 0 \leq \alpha, \beta \leq 2^{L-\mu} - 1 \). Then, the matricisation yields \( U_\mu = \operatorname{span}\{T^{a_{\mu\beta}} : 0 \leq \alpha, \beta \leq 2^{L-\mu} - 1\} \) and \( r_\mu = \dim(U_\mu) \).

A simpler description follows from the fact that

\[
T^{a_{\alpha\beta}} = T \left( [a_{n+(\alpha-\beta)2^\mu}, \ldots, a_{n-2+(\alpha-\beta)2^\mu}] \right) = T(a^{(\alpha-\beta)}),
\]

where \( a^{(\gamma)} = [a_{n+(\gamma-1)2^\mu}, \ldots, a_{n-2+(\gamma+1)2^\mu}] \in \mathbb{R}^{2n+1-1} \) is a part of the vector \( a \) defining \( T = T(a) \). Since the linear map \( a \mapsto T(a) \) is an isomorphism, we obtain the TT-ranks

\[
r_\mu = \dim(U_\mu) = \dim \operatorname{span}\{a^{(\gamma)} : 1 - 2^{L-\mu} \leq \gamma \leq 2^{L-\mu} - 1\}
\]

\[
= \operatorname{rank} \begin{bmatrix}
a_0 & a_2^\mu & \cdots & a_{2^L-2\mu}^\mu \\
a_1 & a_2^{\mu+1} & \cdots & a_{2^L-2\mu+1}^\mu \\
\vdots & \vdots & \ddots & \vdots \\
a_{2^\mu-2} & a_{2^\mu-2}^\mu & \cdots & a_{2^L-2}^\mu
\end{bmatrix}.
\]

(39)

The latter matrix looks similar to the matricisation \( M^{(\mu)} \) in (24). It can be used for the following bound (cf. [14]).
Lemma 5  The TT-rank $r_\mu$ of $T = T(a)$ is bounded by $2r_\mu(a)$, where $r_\mu(a)$ is the TT-rank of the tensorisation of the vector $a \in \mathbb{R}^{2n}$ (here $a_{2n-1}$ can be defined arbitrarily).

Proof  Split the matrix in (39) into the upper part
\[
\begin{bmatrix}
a_0 & \ldots & a_{2n-2\mu} \\
\vdots & \ddots & \vdots \\
a_{2\mu-1} & \ldots & a_{2n-2\mu-1}
\end{bmatrix}
\]
and the lower part
\[
\begin{bmatrix}
a_{2\mu} & \ldots & a_{2n-2\mu} \\
\vdots & \ddots & \vdots \\
a_{2\mu-1} & \ldots & a_{2n-1}
\end{bmatrix},
\]
where the last column is added. The rank (39) is bounded by the sum of the ranks of the latter two matrices. These, however, are submatrices of the matricisation $M^{(\mu)}$ belonging to the vector $a$. This proves the assertion. 

5.3 Matrix-Vector Multiplication

For the evaluation of the product $Tv$, we assume that the Toeplitz matrix $T$ is expressed by the tensorised analogue $T \in \bigotimes_{j=1}^{L} \mathbb{R}^{2 \times 2}$. Here, it is important that for the tensorised quantities $T = \bigotimes_{j=1}^{L} T^{(j)}$ and $y = \bigotimes_{j=1}^{L} y^{(j)}$ the directional product $z := \bigotimes_{j=1}^{L} (T^{(j)} y^{(j)})$ is the tensorisation of $z = Ty$.

The hierarchical representation of $T$ uses the bases $T b^{(\mu)}_{\ell} (1 \leq \ell \leq r_{\mu})$ of $U_{\mu}$, while the leaves $j$ are associated with the subspaces $U_j = U$ spanned by the fixed basis $b^{(1)}_{\mu} := \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $b^{(2)}_{\mu} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and $b^{(3)}_{\mu} := \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$. The coefficient matrices are $T C^{(\mu,\ell)} = \left( T c^{(\mu,\ell)}_{ij} \right)$, i.e.,

\[
T b^{(\mu)}_{\ell} = \sum_{j=1}^{r_{\mu}} \sum_{\ell' = 1}^{3} T c^{(\mu,\ell)}_{ij} b^{(\mu-1)}_{ij} \otimes b^{(\mu)}_{\ell'}.
\]

Let $y \in \mathbb{R}^{n}$ have the tensorised analogue $y \in \bigotimes_{j=1}^{L} \mathbb{R}^{2}$ represented via (26) with data $y_{c_{ij}}^{(\mu+1,\ell)}$ and $y_{b_{ij}}^{(\mu)}$. At the leaves, the basis vectors $b_1 := \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $b_2 := \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ are fixed.

Then, the product $z := Ty \in \mathbb{R}^{2}$ has the tensorised analogue $z \in \bigotimes_{j=1}^{L} \mathbb{R}^{2}$ with data $z_{c_{ij}}^{(\mu+1,\ell)}$ and $z_{b_{ij}}^{(\mu)}$, which are obtained as follows. The recursion

\[
z_{b_{ij}}^{(\mu)} := T b_{\ell}^{(\mu)} y_{b_{ij}}^{(\mu)} = \left( \sum_{i,j} T c^{(\mu,\ell)}_{ij} y_{b_{ij}}^{(\mu-1)} \otimes b^{(\mu)}_{\ell} \right) \left( \sum_{i',j'} y_{c_{i'j'}}^{(\mu,m)} y_{b_{i'j'}}^{(\mu-1)} \otimes b^{(\mu-1)}_{j'} \right)
\]

\[
= \sum_{i,j,i',j'} T c^{(\mu,\ell)}_{ij} y_{c_{i'j'}}^{(\mu,m)} \left( T b_{i}^{(\mu-1)} y_{b_{i}}^{(\mu-1)} \otimes b_{j'} \right) \otimes b_{j'}
\]

\[
= \sum_{i,j} T c^{(\mu,\ell)}_{ij} y_{c_{ij}}^{(\mu,m)} \left( T b_{i}^{(\mu-1)} y_{b_{i}}^{(\mu-1)} \otimes b_{1} \right) + \sum_{i,j} T c^{(\mu,\ell)}_{ij} y_{c_{ij}}^{(\mu,m)} \left( T b_{i}^{(\mu-1)} y_{b_{i}}^{(\mu-1)} \otimes b_{2} \right)
\]

corresponds to (18). Here, we use that at the leaves the products $b_{i}^{(1)} b_{j}$ ($i = 1, 2, 3; j = 1, 2$) are either $b_{1}$ or $b_{2}$ or zero. At the root, we obtain the result $z = Ty = T c^{(L)}_{11} y_{c_{1}^{(L)}}^{(L)} z_{b_{1}}^{(\mu)}$.

The required number of operations is $8 \sum_{\mu=1}^{L} r_{\mu}(T)r_{\mu}(y)r_{\mu-1}(T)r_{\mu-1}(y)$ using Lemma 5 for $T = T(a)$ and the bound $r := \max_{\mu}(r_{\mu}(y), r_{\mu}(a))$, we obtain the work bound $32 \sum_{\mu=1}^{L} r_{\mu}(T)r_{\mu}(y)r_{\mu-1}(T)r_{\mu-1}(y) \leq 32 r^{4}$. Similar to (37), the main cost is required by the orthonormalisation.
6 Additional Remarks

As mentioned above, the convolution can be computed via Fourier forward and backward transforms. As explained in [10, Section 14.4], the Fourier transform \( v \mapsto \hat{v} \) can be realized by using the TT format of the tensorisation of \( v \). The algorithm in Section 4.4 yields the exact convolution. The exact Fourier transform of the tensorized \( v \) may produce intermediate results with increasing rank. Therefore, a statement as in (35d) cannot be obtained. Nevertheless, practical examples with intermediate truncation seem to give satisfactory results (cf. Dolgov et al. [3]).

Funding information Open access funding provided by Max Planck Society.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Braess, D., Hackbusch, W.: On the efficient computation of high-dimensional integrals and the approximation by exponential sums. In: DeVore, R.A., Kunoth, A. (eds.) Multiscale, Nonlinear and Adaptive Approximation, pp. 39–74. Springer, Berlin (2009)
2. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. 21 1253–1278 (2000)
3. Dolgov, S., Khoromskij, B., Savostyanov, D.V.: Superfast Fourier transform using QTT approximation. J. Fourier Anal. Appl 18, 915–953 (2012)
4. Espig, M., Hackbusch, W.: A regularized Newton method for the efficient approximation of tensors represented in the canonical tensor format. Numer. Math. 122, 489–525 (2012)
5. Flad, H.J., Flad Harutyunyan, G.: Singular analysis of RPA diagrams in coupled cluster theory. Manuscript (2017)
6. Grasedyck, L.: Polynomial approximation in hierarchical Tucker format by vector-tensorization. DFG-SPP 1324 Preprint 43 Philipps-Universität Marburg (2010)
7. Hackbusch, W.: Fast and exact projected convolution for non-equidistant grids. Computing 80, 137–168 (2007)
8. Hackbusch, W.: Convolution of hp-functions on locally refined grids.IMA J. Numer. Anal. 29, 960–985 (2009)
9. Hackbusch, W.: Tensorisation of vectors and their efficient convolution. Numer. Math. 119, 465–488 (2011)
10. Hackbusch, W.: Tensor Spaces and Numerical Tensor Calculus. Springer Series in Computational Mathematics, vol. 42. Springer, Berlin (2012)
11. Hackbusch, W.: Hierarchical Matrices: Algorithms and Analysis. Springer Series in Computational Mathematics, vol. 49. Springer, Berlin (2015)
12. Hästad, J.: Tensor rank is NP-complete. J. Algorithms 11, 644–654 (1990)
13. Hitchcock, F.L.: The expression of a tensor or a polyadic as a sum of products. J. Math. Phys. 6, 164–189 (1927)
14. Kazeev, V.A., Khoromskij, B.N., Tyryshnikov, E.E.: Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity. SIAM J. Sci. Comput. 35, A1511–A1536 (2013)
15. Khoromskij, B.: \( O(dN) \)-quantics approximation of \( N \times d \) tensors in high-dimensional numerical modeling. Constr. Approx. 34, 257–280 (2011)
16. Khoromskij, B.N., Veit, A.: Efficient computation of highly oscillatory integrals by using QTT tensor approximation. Comput. Methods Appl. Math. 16, 145–159 (2016)
17. Oseledets, I.V.: Approximation of \( 2^d \times 2^d \) matrices using tensor decomposition. SIAM J. Matrix Anal. Appl. 31, 2130–2145 (2010)
18. Oseledets, I.V., Tyryshnikov, E.E.: Breaking the curse of dimensionality, or how to use SVD in many dimensions. SIAM J. Sci. Comput. 31, 3744–3759 (2009)