Max-Planck-Institut für Mathematik in den Naturwissenschaften Leipzig

Numerical Tensor Techniques for Multidimensional Convolution Products

by

Wolfgang Hackbusch

Preprint no.: 4 2018

Numerical Tensor Techniques for Multidimensional Convolution Products

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

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 $\mathcal{O}(n)$ or larger. The tensorisation method is a representation method introducing additional data-sparsity. In many cases the data size can be reduced from $\mathcal{O}(n)$ to $\mathcal{O}(\log n)$. Even more important, operations as the convolution can be performed with a cost corresponding to these data sizes.

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

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:

$$h(x,z) = \int_{\mathbb{R}^3} f(x, x-y)\, g(y,z)\, \mathrm{d}y \qquad (x, z \in \mathbb{R}^3) \tag{1.1}$$

(for instance, f and g describe the pair amplitude and the pair interaction; cf. Flad–Flad-Harutyunyan [5]). A discretisation by a uniform grid $\{\mathbf{i}h = (i_1 h, i_2 h, i_3 h,) : 0 \le i_1, i_2, i_3 \le n-1\}$ (h: grid size) in a cube leads to the discrete problem

$$h_{\mathbf{ik}} = h^3 \sum\nolimits_{\mathbf{j}} f_{\mathbf{i},\mathbf{i}-\mathbf{j}} g_{\mathbf{j},\mathbf{k}} \qquad (\mathbf{i} = (i_1, i_2, i_3),\ \mathbf{k} = (k_1, k_2, k_3),\ 0 \le i_\nu, k_\nu \le n-1). \tag{1.2}$$

Equation (1.2) describes an unusual matrix multiplication of convolution type:

$$H = F \star G \qquad (H = (h_{\mathbf{ik}}),\ F = (f_{\mathbf{i},\mathbf{j}}),\ G = (g_{\mathbf{j},\mathbf{k}})). \tag{1.3}$$

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 (1.3).

In §2 we shall consider the matrices in (1.3) as tensors of the space[1] $\mathbb{R}^N \otimes \mathbb{R}^N$ with

$$N = n^3. \tag{1.4}$$

Then problem (1.3) reduces to operations of vectors in $\mathbb{R}^N$.

[1]Throughout the paper, $\mathbb{R}$ may be replaced by $\mathbb{C}$.

In a second step (§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* $v \star w$ defined by $(v \star w)_i = \sum_\ell v_{i-\ell} w_\ell$.

The convolution $v \star w$ is a discretisation of the convolution of functions, $\int_\mathbb{R} v(x-y)\,w(y)\,\mathrm{d}y$, 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 §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 $\mathcal{O}(\log^* n)$ data.[2] Then the exact convolution of $v \star w$ requires not more than $\mathcal{O}(\log^* n)$ operations.

The convolution algorithm mentioned above is also interesting outside of quantum chemistry applications. Often, the functions v and w in $\int_\mathbb{R} v(x-y)\,w(y)\,\mathrm{d}y$ 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.[3] 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 $v \star w$.

2 Low-Rank Techniques for Matrices

2.1 Low-Rank Representation

In quantum chemistry it is more usual to write the integral (1.1) as

$$h(x,z) = \int_{\mathbb{R}^3} \tilde{f}(x,y)\,g(y,z)\,\mathrm{d}y \qquad (x,z \in \mathbb{R}^3) \tag{2.1}$$

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 (1.3). However, this notation is less appropriate since the properties of the function f and of the matrix F are swept under the carpet.

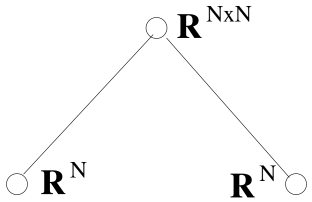

Figure 2.1: Tensor space $\mathbb{R}^N \otimes \mathbb{R}^N \cong \mathbb{R}^{N\times N}$ and its factors $\mathbb{R}^N, \mathbb{R}^N$

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

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

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 (2.1) has infinite rank.

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

$$F = \sum_{\nu=1}^r a_\nu \otimes b_\nu. \tag{2.2}$$

[2] $\log^*(n)$ denotes some (not specified) power of $\log(n)$.
[3] Appropriate algorithms are described in [7], [8].

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 Figure 2.1. In general, the tensor product $\mathbf{v} = v^{(1)} \otimes v^{(2)} \otimes \ldots \otimes v^{(d)}$ with $v^{(j)} \in \mathbb{R}^{n_j}$ is a quantity indexed by d-tuples $\mathbf{i} = (i_1, \ldots, i_d)$ with the values

$$\mathbf{v}[\mathbf{i}] = v^{(1)}[i_1] \cdot v^{(2)}[i_2] \cdot \ldots \cdot v^{(d)}[i_d] \qquad (1 \le i_j \le n_j). \tag{2.3}$$

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

If r is much smaller than N, (2.2) describes the *low-rank representation* of F. Note that the right-hand side of (2.2) requires only $2rN \ll N^2$ data.

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

Remark 2.1 *As a consequence, linear maps on* $\mathbf{V}$ *are uniquely defined by their values of elementary tensors. The same holds for bilinear maps on Cartesian products* $\mathbf{V} \times \mathbf{W}$ *of two tensor spaces.*

2.2 SVD Truncation

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

$$F = \sum\nolimits_{\nu=1}^{r} \sigma_\nu \, a_\nu \otimes b_\nu, \qquad \{a_\nu\}, \{b_\nu\} \text{ orthonormal systems,}$$

with the singular values $\sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_r > 0$. The traditional formulation is $F = U\Sigma V^\mathsf{T}$, where the columns of U and V are defined by a_ν and b_ν, respectively, and Σ is the diagonal matrix containing the singular values.

If $\sigma_{r_\varepsilon} \le \varepsilon$ 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_ε and satisfies the spectral norm estimate $\|F - F_\varepsilon\|_2 \le \varepsilon$.

Now we assume

$$F = \sum_{\nu=1}^{r} a_\nu \otimes b_\nu, \qquad G = \sum_{\mu=1}^{s} c_\mu \otimes d_\mu$$

for the matrices in (1.3). We denote the entries of the vectors $a_\nu, b_\nu, \ldots$ by $a_\nu[\mathbf{i}], b_\nu[\mathbf{i}], \ldots$, where $\mathbf{i}$ abbreviates the triple (i_1, i_2, i_3). Since $F_{\mathbf{i},\mathbf{j}} = \sum_{\nu=1}^{r} a_\nu[\mathbf{i}] b_\nu[\mathbf{j}]$ etc., the operation described in (1.2) becomes

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

$\sum_{\mathbf{j}} b_\nu[\mathbf{i} - \mathbf{j}] \, c_\mu[\mathbf{j}]$ is the component of the convolution $b_\nu \star c_\mu$ at index $\mathbf{i}$. Set $q_{\nu\mu} := b_\nu \star c_\mu$. Then the expression $\sum_{\mathbf{j}} a_\nu[\mathbf{i}] \, b_\nu[\mathbf{i} - \mathbf{j}] \, c_\mu[\mathbf{j}]$ is the $\mathbf{i}$-component of the Hadamard product $a_\nu \odot q_{\nu\mu}$. Together, we obtain the representation of the matrix H in (1.3) by

$$H = \sum_{\mu=1}^{s} \left(h^3 \sum_{\nu=1}^{r} [a_\nu \odot (b_\nu \star c_\mu)] \right) \otimes d_\mu. \tag{2.4}$$

Hence the following has to be calculated:

(a) determine the vectors $q_{\nu\mu} := b_\nu \star c_\mu \in \mathbb{R}^N$,

(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 (2.2). Let

$$U := \operatorname{span}\{a_\nu : 1 \leq \nu \leq r\}, \qquad V := \operatorname{span}\{b_\nu : 1 \leq \nu \leq r\}.$$

Then the tensor (matrix) F satisfies

$$F \in U \otimes V \qquad \text{with } \dim(U) = \dim(V) = r. \tag{2.5}$$

Comparing (2.5) 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 $\{(i_1 h, i_2 h, i_3 h) : 0 \leq i_1, i_2, i_3 \leq n-1\}$. The tensor product of three vectors $a, b, c \in \mathbb{R}^n$ is defined in (2.3). 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 (2.2) would be the representation of $\mathbf{v} \in \mathbf{V} := \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$ by

$$\mathbf{v} = \sum_{\nu=1}^{r} a_\nu \otimes b_\nu \otimes c_\nu. \tag{3.1}$$

The smallest possible value of r is called the *rank* of the tensor $\mathbf{v}$. 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 ≥ 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]).

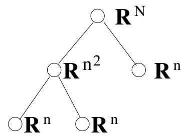

Figure 3.1: Decomposition of $\mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$

The Hadamard product (componentwise product) $\odot$ is a bilinear operation $\mathbf{V} \times \mathbf{V} \to \mathbf{V}$. Another bilinear map is the matrix-vector multiplication. For a unified approach let $\boxdot$ be the symbol of a general bilinear operation between two tensor spaces. An efficient computation of such a tensor operation $\boxdot : \mathbf{X} \times \mathbf{Y} \to \mathbf{Z}$ (with $\mathbf{X} = \bigotimes_{j=1}^d X_j$, etc.) can be based on the following property (3.2), provided this property holds. Let $\mathbf{x} = \bigotimes_{j=1}^d x^{(j)}$ and $\mathbf{y} = \bigotimes_{j=1}^d y^{(j)}$ be elementary tensors[4] with $x^{(j)} \in X_j$, $y^{(j)} \in Y_j$. Then

$$\left(\bigotimes_{j=1}^{d} x^{(j)} \right) \boxdot \left(\bigotimes_{j=1}^{d} y^{(j)} \right) = \bigotimes_{j=1}^{d} \left(x^{(j)} \boxdot_j y^{(j)} \right) \tag{3.2}$$

reduces the operation $\boxdot$ to simpler bilinear operations $\boxdot_j : X_j \times Y_j \to Z_j$ on the individual vector spaces.

In the case of the Hadamard product, $\boxdot = \odot$ is the componentwise product of tensors, while $\boxdot_j = \odot$ is the componentwise product of vectors. In fact, the property

$$(a \otimes b \otimes c) \odot (a' \otimes b' \otimes c') = (a \odot a') \otimes (b \odot b') \otimes (c \odot c') \tag{3.3}$$

follows since $\{(a \otimes b \otimes c) \odot (a' \otimes b' \otimes c')\}[\mathbf{i}] = (a \otimes b \otimes c)[\mathbf{i}] \cdot (a' \otimes b' \otimes c')[\mathbf{i}] = a[i_1]b[i_2]c[i_3]a'[i_1]b'[i_2]c'[i_3]$ and $\{(a \odot a') \otimes (b \odot b') \otimes (c \odot c')\}[\mathbf{i}] = (a \odot a')[i_1]\,(b \odot b')[i_2]\,(c \odot c')[i_3] = a[i_1]a'[i_1]b[i_2]b'[i_2]c[i_3]c'[i_3]$ coincide. Note that on the left-hand side of (3.3) $\odot$ acts on $\mathbf{V} \times \mathbf{V}$, whereas on the right-hand side $\odot$ acts on $\mathbb{R}^n \times \mathbb{R}^n$.

[4] According to Remark 2.1 it is sufficient to investigate the mapping for elementary tensors.

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

$$\left\langle \bigotimes_{j=1}^{d} x^{(j)}, \bigotimes_{j=1}^{d} y^{(j)} \right\rangle = \prod_{j=1}^{d} \left\langle x^{(j)}, y^{(j)} \right\rangle . \tag{3.4}$$

This corresponds to (3.2) with $\mathbf{Y} = \mathbf{X}$ and $\mathbf{Z} = \mathbb{R}$ (the field $\mathbb{R}$ is considered as a tensor space of order $d = 0$).

The notation $(\mathbf{x} \star \mathbf{y})[\mathbf{i}] = \sum_{\mathbf{j}} \mathbf{x}[\mathbf{i} - \mathbf{j}]\, \mathbf{y}[\mathbf{j}]$ of the multivariate convolution involving multiindices $\mathbf{i} \in \mathbb{N}_0^d$ shows that also $\boxdot = \star$ satisfies (3.2). For $d = 3$ we have

$$(a_\nu \otimes b_\nu \otimes c_\nu) \star (a'_\nu \otimes b'_\nu \otimes c'_\nu) = (a_\nu \star a'_\nu) \otimes (b_\nu \star b'_\nu) \otimes (c_\nu \star c'_\nu) . \tag{3.5}$$

Hence, the Hadamard and convolution operations can be reduced to operations acting on vectors in $\mathbb{R}^n$. If $\mathbf{v}$ and $\mathbf{w}$ are given in the form (3.1), all pairs of elementary terms can be treated by (3.3) or (3.5), 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, §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 Figure 3.1.

In the first step we regard the components $v[\mathbf{i}] = v[i_1, i_2, i_3]$ 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 §2 we may write V as $\sum_{\nu=1}^{s} v_\nu^{(12)} \otimes v_\nu^{(3)}$ (cf. (2.2)) with $v_\nu^{(12)} \in \mathbb{R}^{n^2}$ and $v_\nu^{(3)} \in \mathbb{R}^n$. In the second step we regard $v_\nu^{(12)}$ as $n \times n$ matrices or equivalently as tensors of $\mathbb{R}^n \otimes \mathbb{R}^n$ of the form $\sum_{\nu=1}^{r} v_\nu^{(1)} \otimes v_\nu^{(2)}$.

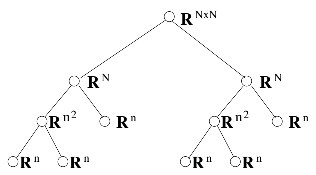

Figure 3.2: Decomposition of $\mathbb{R}^{N \times N}$

Combining the structures of Figures 2.1 and 3.1 yields the splitting depicted in Figure 3.2. 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 (2.5), 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 $U_1, \ldots, U_6 \subset \mathbb{R}^n$. $\mathbf{U}_{12}$ and $\mathbf{U}_{45}$ are subspaces of $\mathbb{R}^n \otimes \mathbb{R}^n \cong \mathbb{R}^{n^2}$, while $\mathbf{U}_{123}$ and $\mathbf{U}_{456}$ are subspaces of $\mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n \cong \mathbb{R}^N$. Also the root $\mathbb{R}^{N \times N}$ has a subspace $\mathbf{U}_{1-6}$. The hierarchical structure is given by

$$\mathbf{U}_\alpha \subset \mathbf{U}_{\alpha_1} \otimes \mathbf{U}_{\alpha_2} \qquad (\alpha_1, \alpha_2 \text{ sons of } \alpha), \tag{3.6}$$

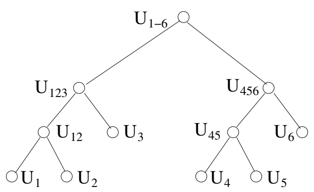

Figure 3.3: Corresponding subspaces

where α belongs to the index set $\{12, 123, 45, 456, \text{1-6}\}$, i.e., $\mathbf{U}_{12} \subset U_1 \otimes U_2$, $\mathbf{U}_{123} \subset \mathbf{U}_{12} \otimes U_3, \ldots,\ \mathbf{U}_{\text{1-6}} \subset \mathbf{U}_{123} \otimes \mathbf{U}_{456}$ (cf. Figure 3.3). The condition (2.5) becomes

$$F \in \mathbf{U}_{\text{1-6}} \qquad \text{(1-6 is the index of the root).} \tag{3.7}$$

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 $\mathbb{R}^{n^2}$, $\mathbb{R}^N = \mathbb{R}^{n^3}$, etc. are too large. Instead we make use of (3.6). Let α be an index of an inner vertex of the tree (no leaf) and α_1, α_2 its sons. Let $\{\mathbf{b}_i^{(\alpha_1)} : 1 \le i \le r_{\alpha_1}\}$ and $\{\mathbf{b}_j^{(\alpha_2)} : 1 \le j \le r_{\alpha_2}\}$ be the bases of $\mathbf{U}_{\alpha_1}$ and $\mathbf{U}_{\alpha_2}$. Then $\{\mathbf{b}_i^{(\alpha_1)} \otimes \mathbf{b}_j^{(\alpha_2)} : 1 \le i \le r_{\alpha_1}, 1 \le j \le r_{\alpha_2}\}$ is a basis of $\mathbf{U}_{\alpha_1} \otimes \mathbf{U}_{\alpha_2}$. A basis vector $\mathbf{b}_\ell^{(\alpha)} \in \mathbf{U}_\alpha \subset \mathbf{U}_{\alpha_1} \otimes \mathbf{U}_{\alpha_2}$ must have a representation

$$\mathbf{b}_\ell^{(\alpha)} = \sum_{i,j} c_{ij}^{(\alpha,\ell)} \mathbf{b}_i^{(\alpha_1)} \otimes \mathbf{b}_j^{(\alpha_2)} \tag{3.8}$$

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

$$C^{(\alpha,\ell)} = (c_{ij}^{(\alpha,\ell)}). \tag{3.9}$$

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

If (3.7) holds, the subspace $\mathbf{U}_{1\text{-}6}$ can be reduced to the one-dimensional space $\mathbf{U}_{\mathrm{root}} = \mathrm{span}\{F\}$. Let $\mathbf{b}_1^{(\mathrm{root})}$ be the only basis vector. Then only one additional factor $c_1^{(\mathrm{root})}$ is needed to characterise

$$F = c_1^{(\mathrm{root})}\mathbf{b}_1^{(\mathrm{root})}. \tag{3.10}$$

Remark 3.1 *(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* $r_{12}r_1r_2 + r_{45}r_4r_5 + r_{123}r_{12}r_3 + r_{456}r_{45}r_6 + 1\cdot r_{123}r_{456}$. $c_1^{(\mathrm{root})}$ *is only one real number. If* $n_j \le n$ *and* $r_j \le r$, *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 (3.6), (3.7) exist and what are their dimensions

$$r_\alpha = \dim(\mathbf{U}_\alpha)$$

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 $F \in \bigotimes_{j=1}^{6}\mathbb{R}^n$ has six indices (we write $F[i_1,\ldots,i_6]$ instead of $F[i_1,i_2,i_3,j_1,j_2,j_3] = F[\mathbf{i},\mathbf{j}]$). The matrix $M^{(45)}$ is of the size $\mathbb{R}^{n^2\times n^4}$ 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

$$\mathbf{U}_{45} := \mathrm{range}(M^{(45)}) \qquad \text{with } r_{45} = \dim(\mathbf{U}_{45}) = \mathrm{rank}(M^{(45)})$$

is the smallest subspace satisfying (3.6), (3.7). For a more general description of the minimal subspaces see [10, §6]

For $\mathbf{v} \in \bigotimes_{j=1}^{d}\mathbb{R}^{n_j}$ let $\emptyset \ne \alpha \subsetneqq \{1,\ldots,d\}$ be a subset with the complement $\alpha^c := \{1,\ldots,d\}\backslash\alpha$. In general, the minimal subspace $\mathbf{U}_\alpha^{\min}(\mathbf{v}) := \mathrm{range}(M^{(\alpha)})$ involves the matricisation $M^{(\alpha)} = M^{(\alpha)}(\mathbf{v})$ 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 α-rank of $\mathbf{v}$ (cf. Hitchcock [13]):

$$\mathrm{rank}_\alpha(\mathbf{v}) := \mathrm{rank}(M^{(\alpha)}(\mathbf{v})).$$

3.4 Hadamard Product and General Bilinear Operations

In the following, the Hadamard product $\odot$ can be replaced by a general bilinear operation $\boxdot$ (cf. (3.2)).

In (2.4) we need the Hadamard product $\mathbf{v}\odot\mathbf{w}$ of two tensors in $\bigotimes_{j=1}^{3}\mathbb{R}^n$. We assume that both $\mathbf{v}$ and $\mathbf{w}$ are represented in the hierarchical format corresponding to the tree depicted in Figure 3.1. $\mathbf{v}$ uses the bases $\{b_i^{(j)} : 1 \le i \le r_j\}, 1 \le j \le 3$, at the leaves and the coefficients $c_{ij}^{(\alpha,\ell)}$, $c_1^{(\mathrm{root})}$, whereas $\mathbf{w}$ is represented by $\{b_i'^{(j)}\}$, $c_{ij}'^{(\alpha,\ell)}$, $c_1'^{(\mathrm{root})}$. Also the ranks r_α and r'_α may be different.

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

$$b''^{(j)}_{(i,i')} := b_i^{(j)} \odot b'^{(j)}_{i'} \qquad (1 \le j \le 3,\ 1 \le i \le r_j,\ 1 \le i' \le r'_j).$$

By induction we assume that the products $\mathbf{b}''^{(\alpha_1)}_{(i,i')}$ and $\mathbf{b}''^{(\alpha_2)}_{(j,j')}$ are (directly or indirectly) determined. Then (3.8) and (3.3) prove that

$$\begin{aligned}\mathbf{b}''^{(\alpha)}_{(\ell,m)} &:= \mathbf{b}^{(\alpha)}_{\ell} \odot \mathbf{b}'^{(\alpha)}_{m} = \left(\sum_{i,j} c^{(\alpha,\ell)}_{ij} \mathbf{b}^{(\alpha_1)}_{i} \otimes \mathbf{b}^{(\alpha_2)}_{j}\right) \odot \left(\sum_{i',j'} c'^{(\alpha,m)}_{i'j'} \mathbf{b}'^{(\alpha_1)}_{i'} \otimes \mathbf{b}'^{(\alpha_2)}_{j'}\right) \\ &= \sum_{i,j}\sum_{i',j'} c^{(\alpha,\ell)}_{ij} c'^{(\alpha,m)}_{i'j'} \left(\mathbf{b}^{(\alpha_1)}_{i} \odot \mathbf{b}'^{(\alpha_1)}_{i'}\right) \otimes \left(\mathbf{b}^{(\alpha_2)}_{j} \odot \mathbf{b}'^{(\alpha_2)}_{j'}\right) \\ &= \sum_{(i,i')}\sum_{(j,j')} c^{(\alpha,\ell)}_{ij} c'^{(\alpha,m)}_{i'j'} \mathbf{b}''^{(\alpha_1)}_{(i,i')} \otimes \mathbf{b}''^{(\alpha_2)}_{(j,j')}.\end{aligned} \tag{3.11}$$

The result $\mathbf{x} := \mathbf{v} \odot \mathbf{w}$ is represented by the generating system $\{b''^{(j)}_{(i,i')}\}$, $1 \le j \le 3$, at the leaves. Here the pairs (i,i') are the indices; thus the index set has the size $r''_j := r_j r'_j$. The equation (3.8) for the new vector contains the coefficients $c''^{(\alpha,(\ell,m))}_{(i,i'),(j,j')} := c^{(\alpha,\ell)}_{ij} c'^{(\alpha,m)}_{i'j'}$. The coefficient $c''^{(\text{root})}_{1}$ is $c^{(\text{root})}_{1} c'^{(\text{root})}_{1}$, since $\mathbf{v} \odot \mathbf{w} = \left(c^{(\text{root})}_{1} \mathbf{b}^{(\text{root})}_{1}\right) \odot \left(c'^{(\text{root})}_{1} \mathbf{b}'^{(\text{root})}_{1}\right) = c^{(\text{root})}_{1} c'^{(\text{root})}_{1} \mathbf{b}^{(\text{root})}_{1} \odot \mathbf{b}'^{(\text{root})}_{1} = c^{(\text{root})}_{1} c'^{(\text{root})}_{1} \mathbf{b}''^{(\text{root})}_{(1,1)}$.

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

Remark 3.2 *The computation of all $b''^{(j)}_{(i,i')}$ requires $3nr_jr'_j$ multiplications. If all coefficients $c''^{(\alpha,(\ell,m))}_{(i,i'),(j,j')}$ are computed explicitly, we need $r_\alpha r'_\alpha r_{\alpha_1} r'_{\alpha_1} r_{\alpha_2} r'_{\alpha_2}$ multiplications. The resulting cost is the product of the data sizes of* $\mathbf{v}$ *and* $\mathbf{w}$.

In §4 the ranks r'_α, r'_{α_1}, r'_{α_2} will be equal to 2.

3.5 Scalar Product, Orthonormalisation, Transformations

As mentioned above, the linear independence of the new frame $\{\mathbf{b}''^{(\alpha)}_{(i,i')}\}$ has to be checked. This can be done by the QR algorithm, provided we are able to determine scalar products $\left\langle \mathbf{b}''^{(j)}_{(i,i')}, \mathbf{b}''^{(j)}_{(m,m')} \right\rangle$ of the vectors determined in (3.11). We simplify the notation (index i instead of (ℓ,m)) and consider the bases $\{\mathbf{b}^{(\alpha)}_{i}\}$ at the vertex α and their connection by (3.8). We proceed from the leaves to the root as in §3.4.

At the leaves the bases are explicitly given so that the scalar products

$$\sigma^{(\alpha)}_{ij} := \left\langle \mathbf{b}^{(\alpha)}_{i}, \mathbf{b}^{(\alpha)}_{j} \right\rangle \tag{3.12}$$

can be determined as usual. As soon as $\sigma^{(\alpha_1)}_{ij}$ and $\sigma^{(\alpha_2)}_{ij}$ are known for the sons of α, $\sigma^{(\alpha)}_{\ell m}$ can be determined by

$$\begin{aligned}\sigma^{(\alpha)}_{\ell m} &= \left\langle \mathbf{b}^{(\alpha)}_{\ell}, \mathbf{b}^{(\alpha)}_{m} \right\rangle = \left\langle \sum_{i,j} c^{(\alpha,\ell)}_{ij} \mathbf{b}^{(\alpha_1)}_{i} \otimes \mathbf{b}^{(\alpha_2)}_{j}, \sum_{i',j'} c^{(\alpha,m)}_{i'j'} \mathbf{b}^{(\alpha_1)}_{i'} \otimes \mathbf{b}^{(\alpha_2)}_{j'} \right\rangle \\ &= \sum_{i,j}\sum_{i',j'} c^{(\alpha,\ell)}_{ij} c^{(\alpha,m)}_{i'j'} \left\langle \mathbf{b}^{(\alpha_1)}_{i}, \mathbf{b}^{(\alpha_1)}_{i'} \right\rangle \left\langle \mathbf{b}^{(\alpha_2)}_{j}, \mathbf{b}^{(\alpha_2)}_{j'} \right\rangle = \sum_{i,j}\sum_{i',j'} c^{(\alpha,\ell)}_{ij} c^{(\alpha,m)}_{i'j'} \sigma^{(\alpha_1)}_{ii'} \sigma^{(\alpha_2)}_{jj'},\end{aligned} \tag{3.13}$$

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 (3.13) terminates at the vertex α, where the scalar products (3.12) are desired.

Of particular interest are orthonormal bases: $\sigma^{(\alpha)}_{ij} = \delta_{ij}$. Using (3.8), we obtain the following result.

Remark 3.3 *Let α be a non-leaf vertex. The basis $\{\mathbf{b}^{(\alpha)}_{\ell}\}$ is orthonormal, if (a) the bases $\{\mathbf{b}^{(\alpha_1)}_{i}\}$ and $\{\mathbf{b}^{(\alpha_2)}_{j}\}$ of the sons α_1, α_2 are orthonormal and (b) the matrices $C^{(\alpha,\ell)}$ in (3.9) are orthonormal with respect to the Frobenius scalar product: $\left\langle C^{(\alpha,\ell)}, C^{(\alpha,m)} \right\rangle_{\mathsf{F}} = \sum_{ij} c^{(\alpha,\ell)}_{ij} c^{(\alpha,m)}_{ij} = \delta_{\ell m}$.*

The bases (or frames) can be orthonormalised as follows. Orthonormalise the explicitly given bases at the leaves (e.g., by QR). As soon as $\{\mathbf{b}_i^{(\alpha_1)}\}$ and $\{\mathbf{b}_j^{(\alpha_2)}\}$ are orthonormal, orthonormalise the matrices $C^{(\alpha,\ell)}$. The new matrices $C_{\text{new}}^{(\alpha,\ell)}$ define a new orthonormal basis $\{\mathbf{b}_{\ell,\text{new}}^{(\alpha)}\}$. 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, §11.3.1.4]).

- Case A1. Let α_1 be the first son of α. Assume that the basis $\{\mathbf{b}_i^{(\alpha_1)}\}$ is transformed into a new basis $\{\mathbf{b}_{i,\text{new}}^{(\alpha_1)}\}$ so that $b_i^{(\alpha_1)} = \sum_k T_{ki}\, \mathbf{b}_{k,\text{new}}^{(\alpha_1)}$. Changing $C^{(\alpha,\ell)}$ into $C_{\text{new}}^{(\alpha,\ell)} := TC^{(\alpha,\ell)}$, the basis $\{\mathbf{b}_\ell^{(\alpha)}\}$ remains unchanged.

- Case A2. If $b_i^{(\alpha_2)} = \sum_k T_{ki}\mathbf{b}_{k,\text{new}}^{(\alpha_2)}$ is a transformation of the second son of α, $C^{(\alpha,\ell)}$ must be changed into $C^{(\alpha,\ell)}T^{\mathsf{T}}$.

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

3.6 SVD Truncation

The example in §3.4 shows that the Hadamard product is given by means of a generating system of increased size $r_j'' := r_j r_j'$. This size may be larger than necessary and should be truncated. The truncation is prepared by an orthonormalisation as described in §3.5.

In principle, the SVD truncation is based on the singular-value decompositions of the matricisations[5] $M^{(\alpha)}$ (cf. §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 $\{\mathbf{b}_{\ell,\text{new}}^{(\alpha)}\}$ together with singular values $\sigma_\ell^{(\alpha)}$ indicating the importance of $\mathbf{b}_{\ell,\text{new}}^{(\alpha)}$. At the start $\alpha = \text{root}$ there is only one (normalised) basis vector $\mathbf{b}_1^{(\text{root})} = \mathbf{b}_{1,\text{new}}^{(\text{root})}$ which remains unchanged. The corresponding weight factor is $\sigma_1^{(\text{root})} = |c_1^{(\text{root})}|$ (cf. (3.10)).

Assume that the new basis $\{\mathbf{b}_{\ell,\text{new}}^{(\alpha)}\}$ is already computed at the vertex α and that α is not a leaf but has sons α_1, α_2. The basis $\{\mathbf{b}_\ell^{(\alpha)}\}$ is characterised by the matrices $C^{(\alpha,\ell)}$. Together with the given values $\sigma_\ell^{(\alpha)}$ we define the matrices[6]

$$\mathbf{Z}_1 := \left[\sigma_1^{(\alpha)}C^{(\alpha,1)}, \sigma_2^{(\alpha)}C^{(\alpha,2)}, \ldots, \sigma_{r_\alpha}^{(\alpha)}C^{(\alpha,r_\alpha)}\right] \in \mathbb{R}^{r_{\alpha_1}\times\left(r_\alpha r_{\alpha_2}\right)},$$
$$\mathbf{Z}_2 := \left[\sigma_1^{(\alpha)}\left(C^{(\alpha,1)}\right)^{\mathsf{T}}, \sigma_2^{(\alpha)}\left(C^{(\alpha,2)}\right)^{\mathsf{T}}, \ldots, \sigma_{r_\alpha}^{(\alpha)}\left(C^{(\alpha,r_\alpha)}\right)^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{\left(r_\alpha r_{\alpha_1}\right)\times r_{\alpha_2}}.$$

The SVD of these matrices yields $\mathbf{Z}_1 = \sum_i \sigma_i^{(\alpha_1)}u_i^{(\alpha_1)}\otimes v_i^{(\alpha_1)}$ and $\mathbf{Z}_2 = \sum_i \sigma_i^{(\alpha_2)}u_i^{(\alpha_2)}\otimes v_i^{(\alpha_2)}$ with orthonormal vectors $u_i^{(\alpha_1)} \in \mathbb{R}^{r_{\alpha_1}}$ and $v_i^{(\alpha_2)} \in \mathbb{R}^{r_{\alpha_2}}$. Now we have to transform the bases at the son nodes: $\{\mathbf{b}_{i,\text{new}}^{(\alpha_1)}\} := \{u_i^{(\alpha_1)}\}$ becomes the new basis for α_1, and $\{\mathbf{b}_{i,\text{new}}^{(\alpha_2)}\} := \{v_i^{(\alpha_2)}\}$ becomes the new basis for α_2. The new bases are called the HOSVD bases (cf. Footnote 5).

The procedure is repeated for the sons of α_1, α_2 until we reach the leaves. Then at all vertices HOSVD bases are introduced together with singular values $\sigma_\nu^{(\alpha)}$. As in §2.2 the SVD truncation consists of omitting all basis vectors corresponding to small enough singular values. Let $\sigma_\nu^{(\alpha)}$, $1 \le \nu \le r_\alpha$, be all singular values at α. Assume that we keep $\sigma_\nu^{(\alpha)}$ for $1 \le \nu \le s_\alpha$ and omit those for $\nu > s_\alpha$. This means that (3.8) is reduced to $\mathbf{b}_\ell^{(\alpha)}$ with $\ell \le s_\alpha$ and that the double sum in (3.8) is taken over $i \le s_{\alpha_1}$ and $j \le s_{\alpha_2}$. Let $\mathbf{v}$ be the input

[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 $\mathbf{Z}_1 = \mathbf{Z}_2$ because $r_{\text{root}} = 1$.

tensor, while $\mathbf{v}_{\mathrm{HOSVD}}$ denotes the truncated version. Then the following estimate holds (cf. [10, Theorem 11.58]):

$$\|\mathbf{v}-\mathbf{v}_{\mathrm{HOSVD}}\| \leq \sqrt{\sum_{\alpha}\sum_{\nu\geq s_\alpha+1}(\sigma_\nu^{(\alpha)})^2} \leq \sqrt{2d-3}\,\|\mathbf{v}-\mathbf{v}_{\mathrm{best}}\|\,. \tag{3.14}$$

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. $\mathbf{v}_{\mathrm{best}}$ is the best approximation with the property that $\mathbf{v}_{\mathrm{best}}$ satisfies $\mathrm{rank}_\alpha(\mathbf{v}_{\mathrm{best}}) \leq s_\alpha$. The parameter d is the order of the tensor, i.e., $d=6$ in the case of Figure 3.2 and $d=3$ for Figure 3.1. Only in the (matrix) case of $d=2$, $\mathbf{v}_{\mathrm{HOSVD}}$ coincides with $\mathbf{v}_{\mathrm{best}}$.

3.7 Convolution

The treatment of §3.4 for the Hadamard operation $\odot$ holds for any binary operation with the property (3.2). Because the multivariate convolution satisfies the analogous condition (3.5), the constructions of §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 $\check{x} = v\star w$ (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). Examples of this technique can be found in Khoromskij [15].[7] Tensorisation 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, etc. The analysis by Grasedyck [6] shows that under suitable conditions, the data size $N(\tilde{\mathbf{v}}_\varepsilon)=\mathcal{O}(\log n)$ 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 $\tilde{\mathbf{v}}_\varepsilon$ and $\tilde{\mathbf{w}}_\varepsilon$. The algorithm developed in §4.4 has a cost related to the data sizes $N(\tilde{\mathbf{v}}_\varepsilon)$, $N(\tilde{\mathbf{w}}_\varepsilon)$.

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 §3 we have to perform various operations with the basis vectors $b_i^{(j)}\in\mathbb{R}^n$. However, more general, the techniques of this chapter can be used for computations in $\mathbb{R}^n$ without connection to the tensor problems in §§2–3.

Tensorisation is an interpretation of an 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 $(i_L, i_{L-1},\ldots,i_1)_2$:

$$k=\sum_{\ell=1}^{L} i_\ell 2^{\ell-1}, \qquad i_\ell\in\{0,1\}. \tag{4.1}$$

[7] In this and later papers the name QTT (quantised TT) is used. We avoid this name since it is inappropriate. The transition from $\mathbb{R}^n$ to $\otimes^L\mathbb{R}^2$ is not a quantisation. Vectors in $\mathbb{R}^n$ are as discrete as tensors in $\otimes^L\mathbb{R}^2$. Another appropriate name would be 'factorisation'.

We map the vector $v \in \mathbb{R}^n$ into the tensor $\mathbf{v} \in \otimes^L \mathbb{R}^2 := \bigotimes_{j=1}^L \mathbb{R}^2$ of order L by means of

$$\mathbf{v}[i_1, \ldots, i_L] = v_k \qquad \text{with } k \text{ and } i_j \text{ as in (4.1).} \tag{4.2}$$

Since $n = \dim(\mathbb{R}^n) = \dim(\otimes^L \mathbb{R}^2) = 2^L$, (4.2) describes an isomorphism

$$\Phi : \otimes^L \mathbb{R}^2 \to \mathbb{R}^n, \quad \mathbf{v} \mapsto v. \tag{4.3}$$

On the side of tensors we shall introduce a hierarchical tensor representation (cf. §3). This allows a simple truncation procedure $\mathbf{v} \mapsto \mathbf{v}_\varepsilon$ (cf. §3.6). Often, the data size $N(\mathbf{v}_\varepsilon)$ of $\mathbf{v}_\varepsilon$ is much smaller than n (see Example 4.4). 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

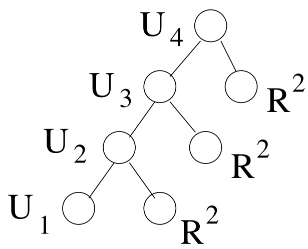

Figure 4.1: Linear tree for the TT format

The underlying tree of the hierarchical representation is the linear tree[8] depicted in Figure 4.1. 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 $b_1^{(j)} = \binom{1}{0}$ and $b_2^{(j)} = \binom{0}{1}$. Figure 4.1 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 $\mathbf{U}_\mu$ belong to $\otimes^\mu \mathbb{R}^2 \cong \mathbb{R}^{2^\mu}$ (in particular $\mathbf{U}_1 = \mathbb{R}^2$).

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

$$M^{(\mu)} = \begin{bmatrix} v_0 & v_{2^\mu} & \ldots & v_{2^{L-1}} \\ v_1 & v_{2^\mu+1} & \ldots & v_{2^{L-1}+1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{2^\mu-1} & v_{2^{\mu+1}-1} & \ldots & v_{2^L-1} \end{bmatrix} \tag{4.4}$$

Since we use the spaces $\mathbb{R}^2$ at the leaves, condition (3.6) becomes

$$\mathbf{U}_{\mu+1} \subset \mathbf{U}_\mu \otimes \mathbb{R}^2 \qquad (1 \le \mu \le L-1), \tag{4.5}$$

while (3.8) is

$$\mathbf{b}_\ell^{(\mu+1)} = \sum_{i=1}^{r_\mu} \left[c_{i1}^{(\mu+1,\ell)} \mathbf{b}_i^{(\mu)} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_{i2}^{(\mu+1,\ell)} \mathbf{b}_i^{(\mu)} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] \quad \text{for } 1 \le \ell \le r_{\mu+1}. \tag{4.6}$$

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

Remark 4.1 *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 $[1/2, 1]$, $[1/4, 1/2]$,..., $[2^{-L}, 2 \cdot 2^{-L}]$, $[0, 2^{-L}]$. 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 $0 \le i \le n-1$. 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 $r_\mu = \operatorname{rank}(M^{(\mu)}) \le g + 2$. Therefore the TT format representing $\mathbf{v} = \Phi^{-1}(F)$ is of the same size as the hp approach. The optimal approximation of f by the TT format with $\operatorname{rank}(M^{(\mu)}) \le g + 2$ 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].*

[8] All binary trees for tensors of order ≤ 3 are linear trees, cf. Figure 3.1.

Example 4.2 *A particular function is the exponential z^x, where $z \neq 0$ may be any complex number. The grid values v_i are ζ^i with $\zeta = z^{2^{-L}}$. For this vector the columns of $M^{(\mu)}$ in (4.4) are linearly dependent so that* $\operatorname{rank}(M^{(\mu)}) = 1$. *In fact, $\mathbf{v} = \Phi^{-1}(v)$ is the elementary tensor $\mathbf{v} = \bigotimes_{j=1}^{L} \begin{pmatrix} 1 \\ \zeta^{2^{j-1}} \end{pmatrix}$. Since* $\sin(ax) = \frac{\exp(\mathrm{i}ax) - \exp(-\mathrm{i}ax)}{2\mathrm{i}}$, *any trigonometric function leads to* $\operatorname{rank}(M^{(\mu)}) = 2$.

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

Remark 4.3 *All functions with a limited number of exponential terms lead to a constant bound of* $\operatorname{rank}(M^{(\mu)})$ *(e.g., $f(x) = \sum_{\nu=1}^{r} \alpha_\nu \exp(-\beta_\nu x)$ yields $\operatorname{rank}(M^{(\mu)}) \leq r$). 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 4.1 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 (1.1) allow a tensorisation with moderate ranks.

Example 4.4 *For $n = 2^L$ set $v = \left(f(k \cdot 2^{-L})\right)_{k=0}^{n-1} \in \mathbb{R}^n$ for the function $f(x) = 1/(1-x)$ in $[0, 1)$. For any $r \in \mathbb{N}$, there is an approximation $v_{(r)} \in \mathbb{R}^n$ such that $\mathbf{v}_{(r)} := \Phi^{-1}(v_{(r)})$ yields ranks $r_\mu = \operatorname{rank}(M^{(\mu)}) \leq r$ and satisfies the componentwise error estimate*

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

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 $\mathbf{v}_{(r)}$ 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 $\sum_{\nu=1}^{r} \alpha_\nu \exp(-\beta_\nu x)$. The error estimates follow from Braess–Hackbusch [1]. ■

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 $\mathbf{v}[i_1, \ldots, i_L] \cdot \mathbf{w}[i_1, \ldots, i_L]$, the property (3.2) holds also in the case of the artificial tensor product $\otimes^L \mathbb{R}^2$; more precisely,

$$\Phi\left(\bigotimes_{j=1}^{L} v^{(j)}\right) \odot \Phi\left(\bigotimes_{j=1}^{L} w^{(j)}\right) = \Phi\left(\bigotimes_{j=1}^{L} \left(v^{(j)} \odot w^{(j)}\right)\right) = \Phi\left(\mathbf{v} \odot \mathbf{w}\right).$$

Conclusion 4.5 *Assume $v = \Phi(\mathbf{v})$ and $w = \Phi(\mathbf{w})$. Let $\mathbf{v}, \mathbf{w}$ be represented by the TT format. Then the Hadamard product $\mathbf{v} \odot \mathbf{w}$ can be computed as explained in §3.4. Since $\Phi(\mathbf{v} \odot \mathbf{w}) = v \odot w$, the result is the tensorisation of $v \odot w$. The computational cost is discussed in §3.4.*

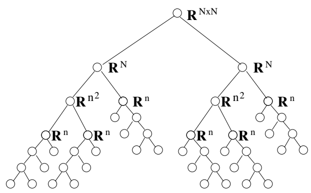

Figure 4.2: Extended tree

We return to the hierarchical format for true tensors as in Figures 3.1 or 3.2. 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 Figure 4.2.

The combination of the tree in Figure 3.1 with the TT tree corresponds to $\mathbb{R}^N \cong \otimes^3 \left(\otimes^L \mathbb{R}^2\right) \cong \otimes^{3L} \mathbb{R}^2$. For tensors represented in this format we can again apply the algorithm in §3.4 to compute $\mathbf{v} \odot \mathbf{w}$ for $\mathbf{v}, \mathbf{w} \in \mathbb{R}^N$.

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 ℓ with $0 \le i - \ell, \ell \le n-1$, i.e.,

$$(v \star w)_i = \sum_{\ell=\max\{0,i+1-n\}}^{\min\{n-1,i\}} v_{i-\ell} w_\ell. \tag{4.7}$$

If i varies in $[0, n-1] \cap \mathbb{Z}$, the sum can be written as $\sum_{\ell=0}^{i}$. For $i < 0$ the empty sum yields $(v \star w)_i = 0$, but for $n \le i \le 2n-2$ the sum in (4.7) is not empty. This shows the following remark.

Remark 4.6 *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 $i \ge n$. The set $\ell_0 = \ell_0(\mathbb{N}_0)$ consists of all sequences with only finitely many nonzero components. Now the sum becomes

$$(v \star w)_i = \sum_{\ell=0}^{i} v_{i-\ell} w_\ell \qquad \text{for all } i \in \mathbb{N}_0 \text{ and all } v, w \in \ell_0. \tag{4.8}$$

Remark 4.7 *The n-periodic convolution is $(v \star_{\text{per}} w)_i = \sum_{\ell=0}^{i} v_{i-\ell} w_\ell \; (0 \le i \le n-1)$, 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 $0 \le i \le n-1$.*

4.4.2 Principal Idea of the Algorithm

For *multivariate* (grid) functions the definition of the convolution implies the property (3.2): 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 $\mathbf{v} \in \otimes^L \mathbb{R}^2$, an obvious question is whether the product of $\mathbf{v} = \otimes_{j=1}^L v^{(j)}$ and $\mathbf{w} = \otimes_{j=1}^L w^{(j)}$ can be expressed by $\mathbf{x} := \otimes_{j=1}^L \left(v^{(j)} \star w^{(j)}\right)$ corresponding to (3.2), i.e., whether the corresponding vectors satisfy $\Phi(\mathbf{v}) \star \Phi(\mathbf{w}) = \Phi(\mathbf{x})$. 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 4.6). Therefore the result does not belong to $\otimes^L \mathbb{R}^2$. Furthermore, we must expect a result in $\otimes^{L+1} \mathbb{R}^2$ since $v \star w$ has the length $2n - 1 > 2^L$ and $< 2^{L+1}$.

4.4.3 Extension to $\otimes^L \ell_0$

According to §4.4.1, $\mathbb{R}^2$ can be considered as a subspace of ℓ_0. Hence $\otimes^L \mathbb{R}^2$ is contained in $\otimes^L \ell_0$. The linear map Φ defined in (4.3) can be extended to $\Phi : \otimes^L \ell_0 \to \ell_0$ by

$$a = \Phi\left(\bigotimes_{j=1}^{L} v^{(j)}\right) \in \ell_0 \qquad \text{with } a_k = \sum_{\substack{i_1,\ldots,i_L \in \mathbb{N}_0 \\ k = \sum_{j=1}^{L} i_j 2^{j-1}}} \prod_{j=1}^{d} v^{(j)}[i_j] \tag{4.9}$$

(cf. Remark 2.1). In the case of $v^{(j)} \in \mathbb{R}^2$, the sum on the right-hand side of (4.9) contains only one term for $0 \le k \le n-1$ and the product $\prod_{j=1}^{L} v^{(j)}[i_j]$ coincides with $\mathbf{v}[i_1, \ldots, i_L]$ for $\mathbf{v} := \bigotimes_{j=1}^{L} v^{(j)}$ (cf. (4.2)).

For a better understanding we look at the case of $L = 2$.

Remark 4.8 *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 $0 \le k < 2i$ and $b_k = a_{k-2i}$ for $k \ge i$.*

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 ℓ_0 and the space $\mathbb{P}$ of polynomials described by

$$\pi : \ell_0 \to \mathbb{P} \quad \text{with } v \mapsto \pi[v](x) := \sum_{k\in\mathbb{N}_0} v_k x^k. \tag{4.10}$$

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

$$\pi[v]\pi[w] = \pi[v \star w] \qquad \text{for } v, w \in \ell_0. \tag{4.11}$$

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

$$\hat{\pi} : \otimes^L \ell_0 \to \mathbb{P} \qquad \text{with } \hat{\pi}\left[\bigotimes_{j=1}^L v^{(j)}\right](x) := \prod_{j=1}^L \pi[v^{(j)}](x^{2^{j-1}}) \tag{4.12}$$

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 4.8 shows that

$$\hat{\pi}\left[\bigotimes_{j=1}^L v^{(j)}\right] = \pi\left[\Phi\left(\bigotimes_{j=1}^L v^{(j)}\right)\right]. \tag{4.13}$$

The extended map $\Phi : \otimes^L \ell_0 \to \ell_0$ is not injective. Two tensors $\mathbf{v}', \mathbf{v}'' \in \otimes^L \ell_0$ are called *equivalent* — denoted by $\mathbf{v}' \sim \mathbf{v}''$ — if they represent the same vector: $\Phi(\mathbf{v}') = \Phi(\mathbf{v}'')$. From (4.13) we learn that the equivalence of $\mathbf{v}', \mathbf{v}''$ can also be expressed by $\hat{\pi}[\mathbf{v}'] = \hat{\pi}[\mathbf{v}'']$.

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

Lemma 4.9 $\Phi\left(\bigotimes\limits_{j=1}^L S^{m_j} v^{(j)}\right) = S^m \Phi\left(\bigotimes\limits_{j=1}^L v^{(j)}\right)$ *holds for* $m = \sum\limits_{j=1}^L m_j 2^{j-1}$.

According to (3.2), we define the convolution of two (elementary) tensors in $\otimes^L \ell_0$ by

$$\left(\bigotimes_{j=1}^L v^{(j)}\right) \star \left(\bigotimes_{j=1}^L w^{(j)}\right) := \bigotimes_{j=1}^L \left(v^{(j)} \star w^{(j)}\right). \tag{4.14}$$

Now the product $v^{(j)} \star w^{(j)}$ makes sense since it belongs to ℓ_0. Next we have to prove that the convolution introduced in (4.14) is consistent with the usual convolution of vectors.

Lemma 4.10 *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* ℓ_0*. Then (4.14) implies*

$$\Phi\left(\bigotimes_{j=1}^L \left(v^{(j)} \star w^{(j)}\right)\right) = v \star w.$$

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

$$\begin{aligned}
\pi\left[\Phi\left(\bigotimes_{j=1}^L \left(v^{(j)} \star w^{(j)}\right)\right)\right](x) &\underset{(4.13)}{=} \hat{\pi}\left[\bigotimes_{j=1}^L \left(v^{(j)} \star w^{(j)}\right)\right](x) \underset{(4.12)}{=} \prod_{j=1}^L \pi[v^{(j)} \star w^{(j)}](x^{2^{j-1}}) \\
&\underset{(4.11)}{=} \prod_{j=1}^L \pi[v^{(j)}](x^{2^{j-1}}) \cdot \pi[w^{(j)}](x^{2^{j-1}}) \\
&= \left(\prod_{j=1}^L \pi[v^{(j)}](x^{2^{j-1}})\right) \cdot \left(\prod_{j=1}^L \pi[w^{(j)}](x^{2^{j-1}})\right) \\
&\underset{(4.12)}{=} \hat{\pi}\left[\bigotimes_{j=1}^L v^{(j)}\right](x) \cdot \hat{\pi}\left[\bigotimes_{j=1}^L w^{(j)}\right](x) \\
&\underset{(4.13)}{=} \pi[v](x) \cdot \pi[w](x) \underset{(4.11)}{=} \pi[v \star w](x).
\end{aligned}$$

■

4.5 Carry-over Procedure

The result $\bigotimes_{j=1}^{L}\left(v^{(j)}\star w^{(j)}\right)$ 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 4.8. 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 4.8, 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 $v\star w$ belongs to $\mathbb{R}^{2n-1}$ the $(L+1)$-th factor must belong to $\mathbb{R}^2$.

4.6 Convolution Algorithm

We recall Remark 4.6: If $\mathbf{v},\mathbf{w}\in\bigotimes_{j=1}^{L}\mathbb{R}^2$, the result is a tensor $\mathbf{u}:=\mathbf{v}\star\mathbf{w}$ in $\bigotimes_{j=1}^{L+1}\mathbb{R}^2$. Lemma 4.11 describes the start at $\delta=1$, while Lemma 4.12 characterises the recursion. In the following the vector notation $v=\left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right]$ means $v_0=\alpha$, $v_1=\beta$, i.e., the components must be read from the top to the bottom. By $\mathbf{v}\sim\mathbf{w}$ we denote the equivalence $\Phi(\mathbf{v})=\Phi(\mathbf{w})$.

Lemma 4.11 *The convolution of* $v=\left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right]$ *and* $w=\left[\begin{smallmatrix}\gamma\\ \delta\end{smallmatrix}\right]\in\mathbb{R}^2=\bigotimes_{j=1}^{1}\mathbb{R}^2$ *yields*

$$\left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right]\star\left[\begin{smallmatrix}\gamma\\ \delta\end{smallmatrix}\right]=\begin{bmatrix}\alpha\gamma\\ \alpha\delta+\beta\gamma\\ \beta\delta\\ 0\end{bmatrix}\sim\left[\begin{smallmatrix}\alpha\gamma\\ \alpha\delta+\beta\gamma\end{smallmatrix}\right]\otimes\left[\begin{smallmatrix}1\\ 0\end{smallmatrix}\right]+\left[\begin{smallmatrix}\beta\delta\\ 0\end{smallmatrix}\right]\otimes\left[\begin{smallmatrix}0\\ 1\end{smallmatrix}\right]\in\bigotimes_{j=1}^{2}\mathbb{R}^2. \tag{4.15a}$$

Furthermore, the shifted vector has the tensor representation

$$S\begin{bmatrix}\alpha\gamma\\ \alpha\delta+\beta\gamma\\ \beta\delta\\ 0\end{bmatrix}=\begin{bmatrix}0\\ \alpha\gamma\\ \alpha\delta+\beta\gamma\\ \beta\delta\end{bmatrix}\sim\left[\begin{smallmatrix}0\\ \alpha\gamma\end{smallmatrix}\right]\otimes\left[\begin{smallmatrix}1\\ 0\end{smallmatrix}\right]+\left[\begin{smallmatrix}\alpha\delta+\beta\gamma\\ \beta\delta\end{smallmatrix}\right]\otimes\left[\begin{smallmatrix}0\\ 1\end{smallmatrix}\right]\in\bigotimes_{j=1}^{2}\mathbb{R}^2. \tag{4.15b}$$

The basic identity is given in the next lemma.

Lemma 4.12 *For given* $\mathbf{v},\mathbf{w}\in\bigotimes_{j=1}^{\delta-1}\mathbb{R}^2$ *let the convolution result be*

$$\mathbf{v}\star\mathbf{w}\sim\mathbf{a}=\mathbf{a}'\otimes\left[\begin{smallmatrix}1\\ 0\end{smallmatrix}\right]+\mathbf{a}''\otimes\left[\begin{smallmatrix}0\\ 1\end{smallmatrix}\right]\in\bigotimes_{j=1}^{\delta}\mathbb{R}^2. \tag{4.16a}$$

Then, convolution of the tensors $\mathbf{v}\otimes x$ *and* $\mathbf{w}\otimes y$ *with* $x=\left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right]$, $y=\left[\begin{smallmatrix}\gamma\\ \delta\end{smallmatrix}\right]\in\mathbb{R}^2$ *yields*

$$\begin{aligned}(\mathbf{v}\otimes x)\star(\mathbf{w}\otimes y)&\sim\mathbf{u}=\mathbf{u}'\otimes\left[\begin{smallmatrix}1\\ 0\end{smallmatrix}\right]+\mathbf{u}''\otimes\left[\begin{smallmatrix}0\\ 1\end{smallmatrix}\right]\in\bigotimes_{j=1}^{\delta+1}\mathbb{R}^2\\ \textit{with}\quad\mathbf{u}'&=\mathbf{a}'\otimes\left[\begin{smallmatrix}\alpha\gamma\\ \alpha\delta+\beta\gamma\end{smallmatrix}\right]+\mathbf{a}''\otimes\left[\begin{smallmatrix}0\\ \alpha\gamma\end{smallmatrix}\right]\in\bigotimes_{j=1}^{\delta}\mathbb{R}^2\\ \textit{and}\quad\mathbf{u}''&=\mathbf{a}'\otimes\left[\begin{smallmatrix}\beta\delta\\ 0\end{smallmatrix}\right]+\mathbf{a}''\otimes\left[\begin{smallmatrix}\alpha\delta+\beta\gamma\\ \beta\delta\end{smallmatrix}\right]\in\bigotimes_{j=1}^{\delta}\mathbb{R}^2.\end{aligned} \tag{4.16b}$$

Proof. Lemma 4.10 implies that

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

Assumption (4.16a) yields

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

Lemma 4.9 shows that

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

Using (4.15a,b), we obtain

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

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

If the vectors x,y in Lemma 4.12 belong to $\left\{\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right],\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right\}$, the vectors $\left[\begin{smallmatrix}\alpha\gamma\\ \alpha\delta+\beta\gamma\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}0\\ \alpha\gamma\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}\beta\delta\\0\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}\alpha\delta+\beta\gamma\\ \beta\delta\end{smallmatrix}\right]$ from (4.16b) belong to $\left\{\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right],\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right],\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right\}$.

Lemma 4.11 proves assumption (4.16a) for $\delta=2$, while Lemma 4.12 shows that $\mathbf{v}\otimes x$ and $\mathbf{w}\otimes y$ satisfy the requirement (4.16a) (for $\delta+1$ instead of δ).

4.7 Convolution of Tensors in Hierarchical Format

We recall that the subspaces $\mathbf{U}_\delta\subset\otimes^\delta\mathbb{R}^2$ satisfy (4.5): $\mathbf{U}_{\delta+1}\subset\mathbf{U}_\delta\otimes\mathbb{R}^2$. The essential observation is that also the results of the convolution yield subspaces with this property.

Note that there are *three* different tensors $\mathbf{v}$, $\mathbf{w}$, $\mathbf{u}:=\mathbf{v}\star\mathbf{w}$ involving representations with three different subspace families $\mathbf{U}'_\delta$, $\mathbf{U}''_\delta$, $\mathbf{U}_\delta$ $(1\le\delta\le L)$. The bases spanning these subspaces consist of the vectors $\mathbf{b}_i'^{(\delta)}$, $\mathbf{b}_i''^{(\delta)}$, $\mathbf{b}_i^{(\delta)}$. The dimensions of the subspaces are r'_δ, r''_δ, r_δ.

Any tensor $\mathbf{a}\in\otimes^\delta\mathbb{R}^2$ $(\delta\ge1)$ can be written as $\mathbf{a}=\mathbf{a}'\otimes\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]+\mathbf{a}''\otimes\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$. Define the linear maps ϕ'_δ, $\phi''_\delta:\otimes^\delta\mathbb{R}^2\to\otimes^{\delta-1}\mathbb{R}^2$ by $\phi'_\delta(\mathbf{a})=\mathbf{a}'$, $\phi''_\delta(\mathbf{a})=\mathbf{a}''$.

Theorem 4.13 *Let the tensors $\mathbf{v},\mathbf{w}\in\bigotimes_{j=1}^L\mathbb{R}^2$ be represented by (possibly different) hierarchical formats using the respective subspaces $\mathbf{U}'_\delta$ and $\mathbf{U}''_\delta$, $1\le\delta\le L$, satisfying*

$$\begin{aligned}\mathbf{U}'_1&=\mathbb{R}^2, &\quad \mathbf{U}'_\delta&\subset\mathbf{U}'_{\delta-1}\otimes\mathbb{R}^2, &\quad \mathbf{v}&\in\mathbf{U}'_L,\\ \mathbf{U}''_1&=\mathbb{R}^2, & \mathbf{U}''_\delta&\subset\mathbf{U}''_{\delta-1}\otimes\mathbb{R}^2, & \mathbf{w}&\in\mathbf{U}''_L.\end{aligned}\tag{4.17a}$$

The subspaces

$$\mathbf{U}_\delta:=\operatorname{span}\{\phi'_{\delta+1}(\mathbf{x}\star\mathbf{y}),\phi''_{\delta+1}(\mathbf{x}\star\mathbf{y}):\mathbf{x}\in\mathbf{U}'_\delta,\ \mathbf{y}\in\mathbf{U}''_\delta\}\qquad(1\le\delta\le L)\tag{4.17b}$$

satisfy

$$\mathbf{U}_1=\mathbb{R}^2,\quad \mathbf{U}_\delta\subset\mathbf{U}_{\delta-1}\otimes\mathbb{R}^2,\quad \mathbf{v}\star\mathbf{w}\in\mathbf{U}_{L+1}.\tag{4.17c}$$

The dimension of $\mathbf{U}_\delta$ can be bounded by

$$\dim(\mathbf{U}_\delta)\le\min\left\{2\dim(\mathbf{U}'_\delta)\dim(\mathbf{U}''_\delta),2^\delta,2^{L+1-\delta}\right\}.\tag{4.17d}$$

Proof. (i) $\mathbf{U}_1=\mathbb{R}^2$ can be concluded from Lemma 4.11.

(ii) Write $\mathbf{x},\mathbf{y}\in\mathbf{U}'_\delta\subset\mathbf{U}'_{\delta-1}\otimes\mathbb{R}^2$ as $\mathbf{x}=\mathbf{x}'\otimes\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]+\mathbf{x}''\otimes\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$ and $\mathbf{y}=\mathbf{y}'\otimes\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]+\mathbf{y}''\otimes\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$ with $\mathbf{x}',\mathbf{x}'',\mathbf{y}',\mathbf{y}''\in\mathbf{U}'_{\delta-1}$. Expansion of the sums yields $\mathbf{x}\star\mathbf{y}=\left(\mathbf{x}'\otimes\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right)\star\left(\mathbf{y}'\otimes\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right)+\ldots$ For each term $\mathbf{z}$ of this expansion, Lemma 4.12 (with $\mathbf{v},\mathbf{w}$ renamed $\mathbf{x}',\mathbf{x}''$) states that $\phi'_{\delta+1}(\mathbf{z})=\mathbf{u}'$ and $\phi''_{\delta+1}(\mathbf{z})=\mathbf{u}''$ belong to $\mathbf{U}_{\delta-1}\otimes\mathbb{R}^2$ (cf. (4.16b)). Hence, $\phi'_{\delta+1}(\mathbf{x}\star\mathbf{y}),\phi''_{\delta+1}(\mathbf{x}\star\mathbf{y})\in\mathbf{U}_{\delta-1}\otimes\mathbb{R}^2$ holds, and the definition of $\mathbf{U}_\delta$ implies the inclusion $\mathbf{U}_\delta\subset\mathbf{U}_{\delta-1}\otimes\mathbb{R}^2$.

(iii) $\mathbf{v}\in\mathbf{U}'_L$ and $\mathbf{w}\in\mathbf{U}''_L$ together with the definition of $\mathbf{U}_L$ lead to $\mathbf{v}\star\mathbf{w}\in\mathbf{U}_L$.

(iv) The first bound of $\dim(\mathbf{U}_\delta)$ follows directly from (4.17b). The bound $\min\{2^\delta, 2^{L+1-\delta}\}$ holds for any $\operatorname{rank}(M^{(1,\dots,\delta)}(\mathbf{v}))$ of $\mathbf{v} \in \otimes^{L+1}\mathbb{R}^2$. ■

The bound $2\dim(\mathbf{U}'_\delta)\dim(\mathbf{U}''_\delta)$ corresponds to the product mentioned in Remark 3.2.

For $\delta = 1, \dots, L$, the numerical scheme has

1. to introduce an orthonormal basis $\{\mathbf{b}_1^{(\delta)}, \dots, \mathbf{b}_{r_\delta}^{(\delta)}\}$ of $\mathbf{U}_\delta$, where $r_\delta := \dim(\mathbf{U}_\delta)$ (cf. §3.5),
2. to represent the convolution $\mathbf{b}_i'^{(\delta)} \star \mathbf{b}_j''^{(\delta)}$ by

$$\mathbf{b}_i'^{(\delta)} \star \mathbf{b}_j''^{(\delta)} = \sum_{k=1}^{r_\delta} \sum_{m=1}^{2} \beta_{ij,km}^{(\delta)} \, \mathbf{b}_k^{(\delta)} \otimes b_m. \tag{4.18}$$

As soon as the β-coefficients from (4.18) are known, general products $\mathbf{x} \star \mathbf{y}$ of $\mathbf{x} \in \mathbf{U}'_\delta$ and $\mathbf{y} \in \mathbf{U}''_\delta$ can be evaluated easily as shown in the next remark.

Remark 4.14 *Let* $\mathbf{x} = \sum_{i=1}^{r'_\delta} \xi_i \mathbf{b}_i'^{(\delta)} \in \mathbf{U}'_\delta$ *and* $\mathbf{y} = \sum_{j=1}^{r''_\delta} \eta_j \mathbf{b}_j''^{(\delta)} \in \mathbf{U}''_\delta$. *Then convolution yields*

$$\mathbf{x} \star \mathbf{y} = \mathbf{z} = \mathbf{z}' \otimes \begin{bmatrix}1\\0\end{bmatrix} + \mathbf{z}'' \otimes \begin{bmatrix}0\\1\end{bmatrix} \quad \textit{with} \quad \mathbf{z}' = \sum_{k=1}^{r_\delta} \zeta_k' \mathbf{b}_k^{(\delta)}, \ \mathbf{z}'' = \sum_{k=1}^{r_\delta} \zeta_k'' \mathbf{b}_k^{(\delta)},$$

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

with $\beta_{ij,km}^{(\delta)}$ *from (4.18). The computation of* ζ_k', ζ_k'' $(1 \le k \le r_\delta)$ *requires* $4 r_\delta r'_\delta (r''_\delta + 1)$ *operations.*

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

$$8 r''_\delta r'_{\delta-1} r_{\delta-1} \left(r''_{\delta-1} + r'_\delta\right) + 8 \left(r'_\delta r''_\delta\right)^2 r_{\delta-1} + \frac{4}{3} \left(r'_\delta r''_\delta\right)^3 + 2 r_{\delta-1} r_\delta^2 \qquad \text{for } 2 \le \delta \le L. \tag{4.19}$$

A rough estimate by $r'_\delta, r''_\delta \le r$ and $r_\delta \le 2r^2$ 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 of $i - j$. A multiplication by a Toeplitz matrix and a convolution are almost equivalent (cf. Kazeev et al. [14]).

If we fix the vector x in $x \star y$, this expression defines a linear map $y \mapsto x \star y$ which may be expressed by a matrix $T = T_x$, i.e., $Ty := x \star y$. In the case of $x, y \in \mathbb{R}^n$ and $x \star y \in \mathbb{R}^{2n-1}$, T is the (rectangular) Toeplitz matrix of size $(2n-1) \times n$ with $T_{i0} = x_i$ $(0 \le i \le n-1)$, $T_{n-1+i,0} = T_{0i} = 0$ $(1 \le i \le n-1)$.

A general $n \times n$ Toeplitz matrix is uniquely determined by the coefficient vector $a = [a_{0,}, \dots, a_{2n-2}]$:

$$T(a) := \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_0 \\ a_n & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-2} \\ a_{2n-2} & \cdots & a_n & a_{n-1} \end{bmatrix}, \qquad \begin{array}{l} \text{i.e., } T(a)_{i,j} = a_{n-1+i-j} \\ \text{for } 0 \le i, j \le n-1. \end{array} \tag{5.1}$$

The product $z := a \star y$ belongs to $\mathbb{R}^{3n-1}$. The part $\hat{z}$ with $\hat{z}_i := z_{n-1+i}$ $(0 \le i \le n-1)$ 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 (4.3) the isomorphism $\mathbf{M} \in \bigotimes_{j=1}^{L}\mathbb{R}^{2\times 2} \mapsto M \in \mathbb{R}^{n\times n}$ is defined by $M[i,j] = \mathbf{M}[(i_1,j_1),\ldots,(i_L,j_L)]$ where $i = \sum_{\ell=1}^{L} i_\ell 2^{\ell-1}$, $j = \sum_{\ell=1}^{L} j_\ell 2^{\ell-1}$, $i_\ell, j_\ell \in \{0,1\}$ (cf. [17]). In particular, a block matrix $\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ corresponds to the tensor product $M_{11} \otimes \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right] + M_{12} \otimes \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right] + M_{21} \otimes \left[\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right] + M_{22} \otimes \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right]$.

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 $b_1 := \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$, $b_2 := \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $b_2 := \left[\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right]$. For the hierarchical representation we use the linear tree of Figure 4.1 with $\mathbb{R}^2$ replaced by U.

The TT-rank $r_\mu = \dim(\mathbf{U}_\mu)$ 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. (5.1)). Consider a regular block structure of T with blocks of size $2^\mu \times 2^\mu$. Denote these blocks by $T^{\alpha\beta} = (T_{ij})_{\alpha 2^\mu \le i \le (\alpha+1)2^\mu - 1,\ \beta 2^\mu \le j \le (\beta+1)2^\mu - 1}$ for $0 \le \alpha, \beta \le 2^{L-\mu} - 1$. Then the matricisation yields $\mathbf{U}_\mu = \operatorname{span}\{T^{\alpha\beta} : 0 \le \alpha,\beta \le 2^{L-\mu} - 1\}$ and $r_\mu = \dim(\mathbf{U}_\mu)$.

A simpler description follows from the fact that

$$T^{\alpha\beta} = T(\left[a_{n+(\alpha-\beta-1)2^\mu}, \cdots, a_{n-2+(\alpha-\beta+1)2^\mu}\right]) = T(a^{(\alpha-\beta)}),$$

where $a^{(\gamma)} = \left[a_{n+(\gamma-1)2^\mu}, \cdots, a_{n-2+(\gamma+1)2^\mu}\right] \in \mathbb{R}^{2^{\mu+1}-1}$ 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

$$\begin{aligned} r_\mu = \dim(\mathbf{U}_\mu) &= \dim\operatorname{span}\{a^{(\gamma)} : 1 - 2^{L-\mu} \le \gamma \le 2^{L-\mu} - 1\} \\ &= \operatorname{rank}\begin{bmatrix} a_0 & a_{2^\mu} & \cdots & a_{2^{2L}-2\cdot 2^\mu} \\ a_1 & a_{2^\mu+1} & \cdots & a_{2^{2L}-2\cdot 2^\mu+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{2\cdot 2^\mu-2} & a_{3\cdot 2^\mu-2} & \cdots & a_{2^{2L}-2} \end{bmatrix}. \end{aligned} \tag{5.2}$$

The latter matrix looks similar to the matricisation $M^{(\mu)}$ in (4.4). It can be used for the following bound (cf. [14]).

Lemma 5.1 *The TT-rank r_μ 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 (5.2) into the upper part $\begin{bmatrix} a_0 & \cdots & a_{2n-2\cdot 2^\mu} \\ \vdots & \ddots & \vdots \\ a_{2^\mu-1} & \cdots & a_{2n-2^\mu-1} \end{bmatrix}$ and the lower part $\begin{bmatrix} a_{2^\mu} & \cdots & a_{2n-2^\mu} \\ \vdots & \ddots & \vdots \\ a_{2\cdot 2^\mu-1} & \cdots & a_{2n-1} \end{bmatrix}$, where the last column is added. The rank (5.2) 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 Ty we assume that the Toeplitz matrix T is expressed by the tensorised analogue $\mathbf{T} \in \bigotimes_{j=1}^{L}\mathbb{R}^{2\times 2}$. Here it is important that for the tensorised quantities $\mathbf{T} = \bigotimes_{j=1}^{L} T^{(j)}$ and $\mathbf{y} = \bigotimes_{j=1}^{L} y^{(j)}$ the directionwise product $\mathbf{z} := \bigotimes_{j=1}^{L}\left(T^{(j)}y^{(j)}\right)$ is the tensorisation of $z = Ty$.

The hierarchical representation of $\mathbf{T}$ uses the bases ${}_T\mathbf{b}_\ell^{(\mu)}$ $(1 \le \ell \le r_\mu)$ of $\mathbf{U}_\mu$, while the leaves j are associated with the subspaces $U_j = U$ spanned by the fixed basis $b_1^U := \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$, $b_2^U := \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $b_3^U := \left[\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right]$. The coefficient matrices are ${}_TC^{(\mu,\ell)} = \left({}_Tc_{ij}^{(\mu,\ell)}\right)$, i.e., ${}_T\mathbf{b}_\ell^{(\mu)} = \sum_{i=1}^{r_\mu}\sum_{j=1}^{3} {}_Tc_{ij}^{(\mu,\ell)}\ {}_T\mathbf{b}_i^{(\mu-1)} \otimes b_j^U$.

Let $y \in \mathbf{R}^n$ have the tensorised analogue $\mathbf{y} \in \bigotimes_{j=1}^L \mathbb{R}^2$ represented via (4.6) with data ${}_yc_{ij}^{(\mu+1,\ell)}$ and ${}_y\mathbf{b}_i^{(\mu)}$. 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 $\mathbf{z} \in \bigotimes_{j=1}^L \mathbb{R}^2$ with data ${}_zc_{(\ell,m),j}^{(\mu+1,\ell)}$ and ${}_z\mathbf{b}_{(\ell,m)}^{(\mu)}$ which are obtained as follows. The recursion

$$\begin{aligned}
{}_z\mathbf{b}_{(\ell,m)}^{(\mu)} &:= {}_T\mathbf{b}_\ell^{(\mu)}\ {}_y\mathbf{b}_m^{(\mu)} = \left(\sum_{i,j} {}_Tc_{ij}^{(\mu,\ell)}\ {}_T\mathbf{b}_i^{(\mu-1)} \otimes b_j^U\right)\left(\sum_{i',j'} {}_yc_{i'j'}^{(\mu,m)}\ {}_y\mathbf{b}_{i'}^{(\mu-1)} \otimes b_{j'}\right) \\
&= \sum_{i,j,i',j'} {}_Tc_{ij}^{(\mu,\ell)}\ {}_yc_{i'j'}^{(\mu,m)} \left({}_T\mathbf{b}_i^{(\mu-1)}\ {}_y\mathbf{b}_{i'}^{(\mu-1)}\right) \otimes \left(b_j^U b_{j'}\right) \\
&= \sum_{i,i'} \sum_{(j,j')\in\{(1,2),(2,1)\}} {}_Tc_{ij}^{(\mu,\ell)}\ {}_yc_{i'j'}^{(\mu,m)} \left({}_T\mathbf{b}_i^{(\mu-1)}\ {}_y\mathbf{b}_{i'}^{(\mu-1)}\right) \otimes b_1 \\
&+ \sum_{i,i'} \sum_{(j,j')\in\{(2,2),(3,1)\}} {}_Tc_{ij}^{(\mu,\ell)}\ {}_yc_{i'j'}^{(\mu,m)} \left({}_T\mathbf{b}_i^{(\mu-1)}\ {}_y\mathbf{b}_{i'}^{(\mu-1)}\right) \otimes b_2
\end{aligned}$$

corresponds to (3.11). Here we use that at the leaves the products $b_i^U b_j$ ($i = 1, 2, 3$; $j = 1, 2$) are either b_1 or b_2 or zero. At the root we obtain the result $\mathbf{z} = \mathbf{Ty} = {}_Tc_1^{(L)}\ {}_yc_1^{(L)}\ {}_z\mathbf{b}_{(1,1)}^{(\mu)}$.

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 5.1 for $T = T(a)$ and the bound $r := \max_\mu\{r_\mu(y), r_\mu(a)\}$, we obtain the work bound $32\sum_{\mu=1}^L r_\mu(T)r_\mu(y)r_{\mu-1}(T)r_{\mu-1}(y) \lesssim 32r^4$. Similar to (4.19) 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, §14.4] the Fourier transform $v \mapsto \hat{v}$ can be realised by using the TT format of the tensorisation of v. The algorithm in §4.4 yields the *exact* convolution. The exact Fourier transform of the tensorised $\mathbf{v}$ may produce intermediate results with increasing rank. Therefore a statement as in (4.17d) cannot be obtained. Nevertheless, practical examples with intermediate truncation seem to give satisfactory results (cf. Dolgov et al. [3]).

References

[1] Braess, D., Hackbusch, W.: On the efficient computation of high-dimensional integrals and the approximation by exponential sums. In: R.A. DeVore, A. Kunoth (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 (2017). Manuscript

[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, *SSCM*, vol. 42. Springer, Berlin (2012)

[11] Hackbusch, W.: Hierarchical Matrices – Algorithms and Analysis, *SSCM*, vol. 49. Springer, Berlin (2015)

[12] Håstad, 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. Journal of Mathematics and Physics **6**, 164–189 (1927)

[14] Kazeev, V.A., Khoromskij, B.N., Tyrtyshnikov, 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.: $O(d \log N)$-quantics approximation of $N-d$ 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., Tyrtyshnikov, E.E.: Breaking the curse of dimensionality, or how to use SVD in many dimensions. SIAM J. Sci. Comput. **31**, 3744–3759 (2009)