Max-Planck-Institut für Mathematik in den Naturwissenschaften Leipzig

Multilevel Toeplitz matrices generated by QTT tensor-structured vectors and convolution with logarithmic complexity

(revised version: August 2011)

by

Vladimir A. Kazeev, Boris N. Khoromskij, and Eugene E. Tyrtyshnikov

Preprint no.: 36 2011

Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity*

Vladimir A. Kazeev† Boris N. Khoromskij‡ Eugene E. Tyrtyshnikov§

August 8, 2011

Abstract

We consider two operations in the QTT format: composition of a multilevel Toeplitz matrix generated by a given multidimensional vector and convolution of two given multidimensional vectors. We show that low-rank QTT structure of the input is preserved in the output and propose efficient algorithms for these operations in the QTT format.

For a d-dimensional $2n \times \ldots \times 2n$-vector $\boldsymbol{x}$ given in a QTT representation with ranks bounded by p we show how a multilevel Toeplitz matrix generated by $\boldsymbol{x}$ can be obtained in the QTT format with ranks bounded by $2p$ in $\mathcal{O}\left(dp^2 \log n\right)$ operations. We also describe how the convolution $\boldsymbol{x} \star \boldsymbol{y}$ of $\boldsymbol{x}$ and a d-dimensional $n \times \ldots \times n$-vector $\boldsymbol{y}$ can be computed in the QTT format with ranks bounded by $2t$ in $\mathcal{O}\left(dt^2 \log n\right)$ operations, provided that the matrix $\boldsymbol{x}\boldsymbol{y}'$ is given in a QTT representation with ranks bounded by t. We exploit approximate matrix-vector multiplication in the QTT format to accelerate the convolution algorithm dramatically.

We demonstrate high performance of the convolution algorithm with numerical examples including computation of the Newton potential of a strong cusp on fine grids with up to $2^{20} \times 2^{20} \times 2^{20}$ points in 3D.

Keywords: Toeplitz matrices, circulant matrices, convolution, tensorisation, virtual levels, tensor decompositions, tensor rank, low-rank representation, Newton potential, Tensor Train, TT, Quantics Tensor Train, QTT.
AMS Subject Classification: 15A69, 15B05, 44A35, 65F99.

1 Introduction

Computation of discrete convolution has been discussed in numerous research articles and monographs. For arbitrary N-component vectors represented elementwise this operation is typically performed by means of the Fast Fourier Transform with the complexity $\mathcal{O}\left(N \log N\right)$, which is unaffordable for large vectors and especially restrictive in high dimensions. Tensor-structured algorithms, which assume that the input data possesses some structure related to separation of variables, i. e. *tensor structure*, provide a dramatic leverage in various computational problems [1, 2, 3], including evaluation of convolution [4, 5].

A novel algorithm *QTT-FFT* of the Fast Fourier Transform in the *Quantics Tensor Train format* [6, 7, 8] with the complexity logarithmic w. r. t. N has been recently proposed and used for computation of discrete convolution [9]. However, the QTT-FFT algorithm requires a lot of TT truncations to

*Partially supported by the RFBR grant 11-01-00549A, the RFBR/DFG grant 09-01-91332, stipends of MPI MiS (Leipzig) and HIM (Bonn), Russian Gov. Contracts II940 and II1112, Priority Research Programs No. 17II and 30OM of RAS.

†Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Str. 8, 119333 Moscow, Russia (vladimir.kazeev@gmail.com). Also a visiting researcher at MPI MiS (Leipzig) and a participant of the Trimester Program on Analysis and Numerics for High Dimensional Problems of HIM (Bonn).

‡Max-Planck-Institut für Mathematik in den Naturwissenschaften, Inselstr. 22-26, D-04103 Leipzig, Germany (bokh@mis.mpg.de). Also a participant of the Trimester Program on Analysis and Numerics for High Dimensional Problems of HIM (Bonn).

§Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Str. 8, 119333 Moscow, Russia (tee@bach.inm.ras.ru). Also a visiting professor of the University of Siedlce, a visiting professor of the University of Chester, a participant of the Trimester Program on Analysis and Numerics for High Dimensional Problems of HIM (Bonn).

be applied to intermediate data that may have much worse QTT structure than the input and output vectors, which impedes application of the algorithm to computation of convolution. Alternatively, we propose a straightforward approach to this particular problem, which yields more advantageous algorithms.

In this paper we study QTT structure of two sorts of objects: the first one is a multilevel Toeplitz matrix $\boldsymbol{T}$ generated by a vector $\boldsymbol{x}$; the second one is a convolution $\boldsymbol{x} \star \boldsymbol{y}$ of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. For the sake of brevity let us consider a circulant $n \times n$-matrix and periodic convolution of n-component vectors in one dimension, which can be written as $\boldsymbol{C}_{ij} \equiv \boldsymbol{x}_{(i-j) \bmod n} = \sum_{m=1}^{n} \boldsymbol{P}_{ijm} \boldsymbol{x}_m$, $1 \le i, j \le n$ and $(\boldsymbol{x} \star \boldsymbol{y})_i \equiv \sum_{j=1}^{n} \boldsymbol{x}_{(i-j) \bmod n} \boldsymbol{y}_j = \sum_{j=1}^{n} \sum_{m=1}^{n} \boldsymbol{P}_{ijm} \boldsymbol{x}_m \boldsymbol{y}_j$, $1 \le i \le n$, where $\boldsymbol{P}$ is a stack of n periodic shift $n \times n$-matrices. We point out that the two operations under consideration are represented in this way in terms of multiplication of the input data (the vector $\boldsymbol{x}$ and the matrix $\boldsymbol{x}\boldsymbol{y}'$) and a "structuring" tensor $\boldsymbol{P}$.

Regarding this special case taken for example, our basic accomplishment is the following: we propose explicitly a rank-2 QTT representation of the tensor $\boldsymbol{P}$. This means that we derive such $U, V \in \mathbb{R}^{2 \times 2 \times 2 \times 2}$ and $W \in \mathbb{R}^{2 \times 2 \times 2 \times 2 \times 2}$ that the following equality holds elementwise for $1 \le i, j, m \le n$:

$$\begin{aligned} \boldsymbol{P}_{ijm} = \sum_{\alpha_{d-1}=1}^{2} \dots \sum_{\alpha_1=1}^{2} & P\left(i_d, j_d, m_d, \alpha_{d-1}\right) \cdot W\left(\alpha_{d-1}, i_{d-1}, j_{d-1}, m_{d-1}, \alpha_{d-2}\right) \cdot \dots \\ & \cdot \; W\left(\alpha_2, i_2, j_2, m_2, \alpha_1\right) \cdot V\left(\alpha_1, i_1, j_1, m_1\right), \end{aligned} \tag{1}$$

where free indices on the right-hand side take values 1 and 2 to represent those on the left-hand side in a binary coding; for example, $i = \overline{i_d \dots i_1} = 1 + \sum_{k=1}^{d} 2^{k-1} (i_k - 1)$. This observation on the QTT structure of $\boldsymbol{P}$ leads to appealing theoretical and practical results. Assume that $\boldsymbol{x}$ and $\boldsymbol{y}$ are given in QTT representations

$$\begin{aligned} \boldsymbol{x}_m = \sum_{\alpha_{d-1}=1}^{p_{d-1}} \dots \sum_{\alpha_1=1}^{p_1} & X_d\left(m_d, \alpha_{d-1}\right) \cdot X_{d-1}\left(\alpha_{d-1}, m_{d-1}, \alpha_{d-2}\right) \cdot \dots \\ & \cdot \; X_2\left(\alpha_2, m_2, \alpha_1\right) \cdot X_1\left(\alpha_1, m_1\right), \quad 1 \le m \le n, \end{aligned} \tag{2}$$

$$\begin{aligned} \boldsymbol{y}_j = \sum_{\beta_{d-1}=1}^{q_{d-1}} \dots \sum_{\beta_1=1}^{q_1} & Y_d\left(j_d, \beta_{d-1}\right) \cdot Y_{d-1}\left(\beta_{d-1}, j_{d-1}, \beta_{d-2}\right) \cdot \dots \\ & \cdot \; Y_2\left(\beta_2, j_2, \beta_1\right) \cdot Y_1\left(\beta_1, j_1\right), \quad 1 \le j \le n, \end{aligned} \tag{3}$$

of ranks $p_{d-1}, \dots, p_1$ and $q_{d-1}, \dots, q_1$ respectively. Then, first, we construct explicitly QTT decompositions of the matrix $\boldsymbol{C}$ with ranks $2p_{d-1}, \dots, 2p_1$; and of the convolution $\boldsymbol{x} \star \boldsymbol{y}$, with ranks $2p_{d-1}q_{d-1}, \dots, 2p_1q_1$. Minimal possible ranks of an exact or ε-accurate QTT decomposition of a tensor are referred to as *QTT ranks* or *ε-ranks of the tensor* [6, 7, 8]. Therefore, our results, in particular, impose upper bounds on QTT ranks or ε-ranks of $\boldsymbol{C}$ and $\boldsymbol{x} \star \boldsymbol{y}$ in terms of those of $\boldsymbol{x}$ and $\boldsymbol{y}$. Furthermore, if $\boldsymbol{x}\boldsymbol{y}'$ is given in a QTT representation

$$\begin{aligned} \left(\boldsymbol{x}\boldsymbol{y}'\right)_{mj} = \sum_{\gamma_{d-1}=1}^{r_{d-1}} \dots \sum_{\gamma_1=1}^{r_1} & G_d\left(m_d, j_d, \gamma_{d-1}\right) \cdot G_{d-1}\left(\gamma_{d-1}, m_{d-1}, j_{d-1}, \gamma_{d-2}\right) \cdot \dots \\ & \cdot \; G_2\left(\gamma_2, m_2, j_2, \gamma_1\right) \cdot G_1\left(\gamma_1, m_1, j_1\right), \quad 1 \le m, j \le n, \end{aligned} \tag{4}$$

of ranks $r_{d-1}, \dots, r_1$, then $\boldsymbol{x} \star \boldsymbol{y}$ has a QTT decomposition with ranks $2r_{d-1}, \dots, 2r_1$. This yields better estimates of the ranks when $r_{d-1}, \dots, r_1$ are smaller than $p_{d-1}q_{d-1}, \dots, p_1q_1$, which means that the vectors convolved have some structure in common and is typically the case when we deal with discretizations of reasonable problems.

Second, we propose practical algorithms computing $\boldsymbol{C}$ and $\boldsymbol{x} \star \boldsymbol{y}$ as the decompositions constructed explicitly or their approximations, based on matrix-vector multiplication in the QTT format. How to perform the latter basic operation efficiently is a general and fundamental question, but, no

matter what particular method is used, its complexity depends drastically on ranks of the QTT decompositions inputed and arising in computations. Fortunately, in many practical situations we may expect ranks to be low, i. e. about 10, tens or $\mathcal{O}(d)$ [10, 11]. If we let all QTT ranks of decompositions of $\boldsymbol{x}$ (2) and $\boldsymbol{y}$ (3) equal to p and q respectively, then the explicit exact multiplications computing $\boldsymbol{C}$ and $\boldsymbol{x} \star \boldsymbol{y}$ cost $\mathcal{O}\left(p^2 \log n\right)$ and $\mathcal{O}\left(p^2 q^2 \log n\right)$ flops respectively. In some cases even this straightforward approach preforms well enough, while more sophisticated approximate multiplication methods with on-the-fly truncation may perform far much better (see Section 5).

All the results presented above briefly in the case of circulant matrices and periodic convolution in one dimension are obtained in this paper for multilevel Toeplitz matrices and a few important types of convolution in many dimensions, complexity of the algorithms scaling linearly w. r. t. the number of dimensions.

It is also to be pointed out that the representation (1) and further results were obtained with the use of the technique developed in [12], where QTT structure of matrices was studied analytically for the first time and explicit QTT decompositions of some matrices, including the Laplace operator in D dimensions and, in one dimension, its inverse as well, were presented.

1.1 Bibliography overview

Toeplitz matrices are widely used in mathematics, physics and engineering. The problems they arise in, such as solution of integral and partial differential equations, signal and image processing, queuing problems and time series analysis, exploit various Toeplitz-specific computational algorithms for matrix-vector multiplication, linear system preconditioning and solution, matrix inversion and the eigenvalue problem [13, 14, 15, 16]. Toeplitz structure was also generalized to the *displacement structure* possessed by *Toeplitz-like matrices* [17]. However, the Toeplitz (as well as Toeplitz-like) structure itself does not allow one to gain the asymptotic complexity sublinear w. r. t. the matrix size.

Another kind of structure brought into play in order to reduce the complexity of computations with Toeplitz matrices is the *tensor structure*, which is related to the idea of separation of variables for the sake of low-parametric data representation and handling. Several tensor decompositions generalizing the low-rank representation of matrices or, to refer to algorithmic aspects, the Singular Value Decomposition of them, are presented in the surveys [1, 2, 3]. The one that has been most extensively employed to Toeplitz matrices and convolution so far is the *canonical decomposition*, also known as *CANDECOMP*, *PARAFAC* and *CP*.

CP structure of multilevel Toeplitz matrices related to that of generating vectors was presented in [18] and a general approach to fast algorithms for multilevel tensor-structured matrices in the CP format was considered in the same paper. For two-level Toeplitz matrices a fast approximate CP-structured inversion algorithm with the complexity typically sublinear w. r. t. the matrix size was introduced in [19].

On the other hand, in scientific computing much more emphasis has been placed so far on CP-structured methods of convolution, which may alleviate evaluation of potentials in 3D drastically compared to the FFT-based convolution algorithm in the full format . These methods rely on reduction of computation of the multidimensional convolution $\boldsymbol{z}_{i_1,\ldots,i_D} = \sum_{\gamma=1}^{r} \prod_{K=1}^{D} Z_K(i_K, \gamma)$, where $i_K = 1, \ldots, n$ for $K = 1, \ldots, D$, of two D-dimensional vectors given in the canonical representations $\boldsymbol{x}_{m_1,\ldots,m_D} = \sum_{\alpha=1}^{p} \prod_{K=1}^{D} X_K(m_K, \alpha)$ and $\boldsymbol{y}_{j_1,\ldots,j_D} = \sum_{\beta=1}^{q} \prod_{K=1}^{D} Y_K(j_K, \beta)$, where $m_K, j_K = 1, \ldots, n$ for $K = 1, \ldots, D$, of ranks p and q respectively to evaluation of $D \cdot p \cdot q$ one-dimensional full-format convolutions

$$Z_K(\cdot, \alpha\beta) = X_K(\cdot, \alpha) \star Y_K(\cdot, \beta) \tag{5}$$

of their canonical factors, so that $\gamma = (\alpha, \beta)$ and $r = p \cdot q$ (see, for example, [20]). While the basic idea is simple, a truncation procedure needed to wind up with moderate ranks, say $\mathcal{O}\left(\sqrt{pq}\right)$ instead of $\mathcal{O}(pq)$, is to be considered as a necessary ingredient yielding a particular method. Computation of the exact representation (5) requires $\mathcal{O}(Dpq\, n \log n)$ operations and, thus, has the complexity sublinear w. r. t. the problem size n^D, provided that the canonical ranks of the input vectors are

small enough. This makes the CP-structured convolution more favorable compared to the FFT-based convolution in the full format, for example, in quantum chemistry computations [21, 22, 23, 20, 24, 25] and other applications (see the paper [5] on linear filtering and references therein) and allows (up to efficiency of the truncation procedure) to avoid the "curse of dimensionality" [26]. We would like also to mention the method combining CP-structured approach to convolution with local grid refinement strategies, that was proposed in [27] and developed further in [28].

However, the CP-structured convolution has two major downsides. First, there is no robust truncation procedure in the CP format available and, furthermore, such a procedure cannot exist since the format itself is unstable and the best rank-r approximation problem can easily turn out to be ill-posed for $r > 1$ [29]. The *Tensor Train (TT) decomposition* [30, 31, 32, 33, **?**], which we employ in this paper for tensor-structured representation of multilevel Toeplitz matrices and multidimensional convolution, is free from this disadvantage and is equipped with a robust arithmetics including truncation. It is to be pointed out that the TT format, in fact, has been known and exploited for almost two decades now as the *Matrix Product States (MPS)* underlying the *Density Matrix Renormalization Group (DMRG)* approach to quantum spin systems proposed by White in 1992 and widely used by physicists nowadays [34, 35, 36]. The same concept was also introduced in the quantum information theory by Vidal in 2003 as the *state decomposition* [37]. In this paper we use the CP structure of convolution, exploited in (5), in the TT format the same straightforwardly.

But the main point of this paper relates more to the second drawback of the CP-structured convolution, which is that the complexity is still higher than linear w. r. t. n. In order to make it sublinear, one may adopt the *QTT format* [6, 7, 8], which is the particular case of the TT format with the smallest possible mode sizes, applied to tensors reshaped correspondingly. This allows to propose methods with the complexity logarithmic w. r. t. n. *QTT-FFT*, an FFT algorithm in the QTT format attaining such a complexity, was recently proposed and shown to be an efficient tool for fast evaluation of convolution [9]. The similar transformation of the tensors could be coupled formally with CP instead of TT, but due to restrictiveness of the former and the lack of robust arithmetics for it this would make the problem even less tractable than in the case of the "regular" CP.

The idea of introducing additional dimensions (virtual levels) was applied to analysis of canonical decomposition of asymptotically smooth functions as early as in 2003 in [38]. In that paper ranks of particular unfoldings, which are now referred to as *QTT ranks* since [8, 7], were estimated above. After the TT format, the idea of "tensorization" of vectors was adopted for the Hierarchical Tucker format [39] in [40]. Algebraic view of convolution of tensorized vectors in the Hierarchical Tucker format was presented in [41].

The concept of *TT ranks* is crucial for the present paper. According to the basic paper on TT [32], the $D-1$ TT ranks of a D-dimensional tensor are the ranks of the corresponding unfoldings $\boldsymbol{X}^{(k)}$ of $\boldsymbol{x}$, which are obtained from $\boldsymbol{x}$ by reshaping, indices $1, \ldots, k$ and $k+1, \ldots, d$ being considered as row and column indices respectively: $\boldsymbol{X}^{(k)}{}_{i_1 \ldots i_k;\, i_{k+1} \ldots i_D} = \boldsymbol{x}_{i_1 \ldots i_d}$. The TT ranks of a tensor defined in such a way are the minimal possible ranks of an exact TT decomposition of the tensor. They are important in view of storage costs and complexity of the TT arithmetics operations, such as the dot product, multidimensional contraction, matrix-vector multiplication, rank reduction and orthogonalization of a decomposition are polynomial w. r. t. TT ranks of the tensor involved [33]. *QTT ranks* of a tensor are defined as the TT ranks of the tensor subject to a proper reshaping. For example, QTT ranks of the decomposition given by (2) are $p_{d-1}, \ldots, p_1$.

2 Notation

Some tensor notation. We use the symbol "|" to denote three-dimensional tensors by listing their slices along the third mode. To put it specifically, if A_{i_3}, $1 \le i_3 \le n_3$, are $n_1 \times n_2$-matrices, then by $A_1|A_2|\ldots|A_{n_3}$ we mean a $n_1 \times n_2 \times n_3$-tensor with elements $A_{i_1 i_2 i_3} = (A_{i_3})_{i_1 i_2}$.

For tensor contraction of tensors A and B we use the notation $A \bullet_{\eta_1, \ldots, \eta_d}^{\xi_1, \ldots, \xi_d} B$, writing the corresponding modes of A and B they are to be summed over on the top and at the bottom of the

contraction mark, respectively. By omitting contraction modes of a tensor we mean that all modes of the tensor are contracted. For example, for matrices A and B and a vector x this implies that $A \bullet_1^2 B = AB$, $A \bullet_2^2 B = AB'$, $A \bullet_{1,2}^{1,2} B = A \bullet B = \sum_{i,j} A_{ij} B_{ij} = \langle A, B \rangle$ and $A \bullet_1^2 x = A \bullet^2 x = Ax$.

We also extensively use the following binary representation of indices: by $i = \overline{i_d \dots i_1} = 1 + \sum_{k=1}^{d} 2^{k-1} (i_k - 1)$ we mean a scalar index with the range $1, \dots, 2^d$, while the scalar binary indices i_k, $1 \le k \le d$, take values 1 and 2.

Core matrices and core products. By a TT *core* of rank $p \times q$ and mode size $n_1 \times \dots \times n_\nu$ we mean a $\nu + 2$-dimensional array with two *rank indices* varied in the ranges $1, \dots, p$ and $1, \dots, q$ and ν *mode indices* varied in the ranges $1, \dots, n_\kappa$, $1 \le \kappa \le \nu$. We refer to subarrays of a core, obtained by fixing both the rank indices, as *blocks* of the core. In order to focus on rank structure of a core we may consider it as a matrix, indexed by the two rank indices, with entries that are blocks of the core. We call such matrices *core matrices*.

For example, let $n_1 \times \dots \times n_\nu$-tensors $A_{\alpha\beta}$, $\alpha = 1, \dots, p$, $\beta = 1, \dots, q$ be blocks of a core U of rank $p \times q$ and mode size $n_1 \times \dots \times n_\nu$, i. e. $U(\alpha, i_1, \dots, i_\nu, \beta) = (A_{\alpha\beta})_{i_1 \dots i_\nu}$ for all values of the indices involved. Then we write the core matrix of U as

$$U = \begin{bmatrix} A_{11} & \cdots & A_{1q} \\ \vdots & \vdots & \vdots \\ A_{p1} & \cdots & A_{pq} \end{bmatrix}. \tag{6}$$

In order to avoid confusion we use parentheses for regular matrices, which are to be multiplied as usual, and square brackets for cores (core matrices), which are to be multiplied by means of the two core products "$\bowtie$" and "$\bullet$" introduced in [12] and defined below. Addition of cores is meant elementwise, as well as that of matrices or tensors. Any $n_1 \times \dots \times n_\nu$-tensor A can be regarded as a core of rank 1×1. Then the core products defined below coincide with corresponding operations over tensors. Also we may think of $A_{\alpha\beta}$ or any submatrix of the core matrix in (6) as of *subcores* of U.

Definition 2.1 (Rank core product)**.** Consider cores U_1 and U_2 of ranks $r_0 \times r_1$ and $r_1 \times r_2$, composed of blocks $A^{(1)}_{\alpha_0\alpha_1}$ and $A^{(2)}_{\alpha_1\alpha_2}$, $1 \le \alpha_k \le r_k$ for $0 \le k \le 2$, of mode sizes $n_1^{(1)} \times \dots \times n_\nu^{(1)}$ and $n_1^{(2)} \times \dots \times n_\nu^{(2)}$ respectively. Let us define a *rank product* $U_1 \bowtie U_2$ of U_1 and U_2 as a core of rank $r_0 \times r_2$, consisting of blocks

$$A_{\alpha_0\alpha_2} = \sum_{\alpha_1=1}^{r_1} A^{(1)}_{\alpha_0\alpha_1} \otimes A^{(2)}_{\alpha_1\alpha_2}, \quad 1 \le \alpha_0 \le r_0, \quad 1 \le \alpha_2 \le r_2,$$

of mode size $n_1^{(1)} n_1^{(2)} \times \dots \times n_\nu^{(1)} n_\nu^{(2)}$.

In other words, we define $U_1 \bowtie U_2$ as a regular matrix product of the two corresponding core matrices, their elements (blocks) being multiplied by means of tensor product. For example,

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \bowtie \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11} \otimes B_{11} + A_{12} \otimes B_{21} & A_{11} \otimes B_{12} + A_{12} \otimes B_{22} \\ A_{21} \otimes B_{11} + A_{22} \otimes B_{21} & A_{21} \otimes B_{12} + A_{22} \otimes B_{22} \end{bmatrix}.$$

Definition 2.2 (Mode core product)**.** Consider TT cores U_1 and V_1 of ranks $p_0 \times p_1$ and $q_0 \times q_1$, composed of blocks $A^{(1)}_{\alpha_0\alpha_1}$, $1 \le \alpha_0 \le p_0$, $1 \le \alpha_1 \le p_1$, and $B^{(1)}_{\beta 0 \beta_1}$, $1 \le \beta_0 \le q_0$, $1 \le \beta_1 \le q_1$ respectively. Let us define a *mode product* $U_1 \bullet_{\eta_1,\dots,\eta_d}^{\xi_1,\dots,\xi_d} V_1$ of U_1 and V_1 over d modes $\xi_1, \dots, \xi_d$ of U_1 and d modes $\eta_1, \dots, \eta_d$ of V_1 as a core of rank $p_0 q_0 \times p_1 q_1$, consisting of blocks

$$C^{(1)}_{\alpha_0\beta_0;\alpha_1\beta_1} = A^{(1)}_{\alpha_0\alpha_1} \bullet_{\eta_1,\dots,\eta_d}^{\xi_1,\dots,\xi_d} B^{(1)}_{\beta_0\beta_1}, \quad 1 \le \alpha_\kappa \le p_\kappa, \quad 1 \le \beta_\kappa \le q_\kappa, \quad \kappa = 0, 1.$$

This definition implies that we consider a tensor product of corresponding core matrices, their elements (blocks) being multiplied by means of tensor contraction w. r. t. the specified modes. Similarly to tensors, when a core is involved in the mode core product operation w. r. t. all its modes,

for the sake of brevity we omit their list at the corresponding position near the symbol "$\bullet$". For instance, for matrices $A_{\alpha\beta}$ and vectors $X_{\alpha\beta}$ we can write

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \bullet^2 \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} A_{11}X_{11} & A_{11}X_{12} & A_{12}X_{11} & A_{12}X_{12} \\ A_{11}X_{21} & A_{11}X_{22} & A_{12}X_{21} & A_{12}X_{22} \\ A_{21}X_{11} & A_{21}X_{12} & A_{22}X_{11} & A_{22}X_{12} \\ A_{21}X_{21} & A_{21}X_{22} & A_{22}X_{21} & A_{22}X_{22} \end{bmatrix}.$$

The two core products are similar, tensor contraction and tensor product being interchanged in them. The core products inherit some basic properties of tensor contraction (in particular, regular matrix multiplication) and tensor product, which we employ routinely throughout the paper. For instance, we can transform rows and columns of core matrices just the same way as we do it with regular matrices:

$$\begin{aligned} \begin{bmatrix} \alpha_1 U_1 & \beta_1 U_1 \\ \alpha_1 V_1 & \beta_1 V_1 \end{bmatrix} \bowtie \begin{bmatrix} \alpha_2 U_2 & \alpha_2 V_2 \\ \beta_2 U_2 & \beta_2 V_2 \end{bmatrix} &= \left(\begin{bmatrix} U_1 \\ V_1 \end{bmatrix} \bowtie \begin{bmatrix} \alpha_1 & \beta_1 \end{bmatrix} \right) \bowtie \left(\begin{bmatrix} \alpha_2 \\ \beta_2 \end{bmatrix} \bowtie \begin{bmatrix} U_2 & V_2 \end{bmatrix} \right) \\ &= \begin{bmatrix} U_1 \\ V_1 \end{bmatrix} \bowtie \left(\begin{bmatrix} \alpha_1 & \beta_1 \end{bmatrix} \bowtie \begin{bmatrix} \alpha_2 \\ \beta_2 \end{bmatrix} \right) \bowtie \begin{bmatrix} U_2 & V_2 \end{bmatrix} \\ &= (\alpha_1\alpha_2 + \beta_1\beta_2) \begin{bmatrix} U_1 \\ V_1 \end{bmatrix} \bowtie \begin{bmatrix} U_2 & V_2 \end{bmatrix} \end{aligned} \tag{7}$$

for any coefficients $\alpha_1, \beta_1, \alpha_2, \beta_2$ and blocks or subcores U_1, V_1, U_2, V_2 of proper ranks and mode sizes. The two core products introduced above are helpful in dealing with TT decompositions. For example, (1) and (2) can be recast as $\boldsymbol{P} = P \bowtie W \bowtie \ldots \bowtie W \bowtie V$ and $\boldsymbol{x} = X_d \bowtie X_{d-1} \bowtie \ldots \bowtie X_2 \bowtie X_1$.

Let $\boldsymbol{A} = U_d \bowtie \ldots \bowtie U_1$ and $\boldsymbol{B} = V_d \bowtie \ldots \bowtie V_1$, then a linear combination of $\boldsymbol{A}$ and $\boldsymbol{B}$ can be put down in the following way:

$$\alpha\boldsymbol{A} + \beta\boldsymbol{B} = \begin{bmatrix} U_d & V_d \end{bmatrix} \bowtie \begin{bmatrix} U_{d-1} & \\ & V_{d-1} \end{bmatrix} \bowtie \ldots \bowtie \begin{bmatrix} U_2 & \\ & V_2 \end{bmatrix} \bowtie \begin{bmatrix} \alpha U_1 \\ \beta V_1 \end{bmatrix};$$

a tensor product of $\boldsymbol{A}$ and $\boldsymbol{B}$, as $\boldsymbol{A} \otimes \boldsymbol{B} = U_d \bowtie \ldots \bowtie U_1 \bowtie V_d \bowtie \ldots \bowtie V_1$; a transpose $\boldsymbol{A}'$ of $\boldsymbol{A}$ is equal to the rank core product of the same cores, their blocks being transposed; a Frobenius product of $\boldsymbol{A}$ and $\boldsymbol{B}$ is $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{ij} \boldsymbol{A}_{ij} \boldsymbol{B}_{ij} = (U_d \bullet V_d) \bowtie \ldots \bowtie (U_1 \bullet V_1)$; a matrix product of $\boldsymbol{A}$ and $\boldsymbol{B}$ and a matrix-vector product of $\boldsymbol{A}$ and $\boldsymbol{x} = X_d \bowtie X_{d-1} \bowtie \ldots \bowtie X_2 \bowtie X_1$, as $\boldsymbol{AB} = \boldsymbol{A} \bullet_1^2 \boldsymbol{B} = \left(U_d \bullet_1^2 V_d\right) \bowtie \ldots \bowtie \left(U_1 \bullet_1^2 V_1\right)$ and $\boldsymbol{Ax} = \boldsymbol{A} \bullet^2 \boldsymbol{x} = \left(U_d \bullet^2 X_d\right) \bowtie \ldots \bowtie \left(U_1 \bullet^2 X_1\right)$ respectively. The latter equalities can be trivially generalized to the case of the mode product of TT cores presented by Definition 2.2.

Proposition 2.3. *Mode product of two rank products can be recast core-wise:*

$$(U_1 \bowtie U_2) \bullet_{\eta_1,\ldots,\eta_d}^{\xi_1,\ldots,\xi_d} (V_1 \bowtie V_2) = \left(U_1 \bullet_{\eta_1,\ldots,\eta_d}^{\xi_1,\ldots,\xi_d} V_1\right) \bowtie \left(U_2 \bullet_{\eta_1,\ldots,\eta_d}^{\xi_1,\ldots,\xi_d} V_2\right).$$

The core notation introduced just above bears a strong resemblance to the MPS notation (see Section 1.1), according to which, for instance, the right-hand side of (2) is usually written as the matrix product $X_d^{(m_d)} \cdot X_{d-1}^{(m_{d-1})} \cdot \ldots \cdot X_2^{(m_2)} \cdot X_1^{(m_1)}$ of a row, $d-2$ matrices and a column indexed by rank indices $\alpha_{d-1}, \ldots, \alpha_1$ and depending also on mode indices as parameters. In our calculations we prefer to omit the mode indices correctly, so neither the MPS notation nor the elementwise one of (2) is convenient enough for our purposes and we have to use a more suitable core notation.

Elementary QTT blocks. We will describe QTT structure of tensors in terms of the following four TT blocks of size 2×2:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad J = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad J' = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Other notation. Finally, by $A^{\otimes k}$, k being nonnegative integer, we mean a k-th tensor power of A. For example, $I^{\otimes 3} = I \otimes I \otimes I$, and likewise for the rank core product operation "$\bowtie$".

3 QTT structure of shift matrices

3.1 One dimension

Let us consider the following one-dimensional shift $2^d \times 2^d$-matrices:

$$\boldsymbol{P}_2^{(d)} = \begin{pmatrix} 0 & & & & 1 \\ 1 & \ddots & & & \\ 0 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & 1 & 0 \end{pmatrix}$$

of periodic downward shift and

$$\boldsymbol{Q}_2^{(d)} = \begin{pmatrix} 0 & & & \\ 1 & \ddots & & \\ 0 & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ & & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{R}_{2^d}^{(d)} = \begin{pmatrix} 0 & 1 & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 0 \\ & & & \ddots & 1 \\ & & & & 0 \end{pmatrix}$$

of zero fill downward and upward shift respectively.

Lemma 3.1. *The following QTT representations of the shift matrices hold.*

$$\begin{aligned} \boldsymbol{P}_2^{(d)} &= \begin{bmatrix} I & P \end{bmatrix} \bowtie \begin{bmatrix} I & J' \\ & J \end{bmatrix}^{\bowtie(d-2)} \bowtie \begin{bmatrix} J' \\ J \end{bmatrix}, \\ \boldsymbol{Q}_2^{(d)} &= \begin{bmatrix} I & J' \end{bmatrix} \bowtie \begin{bmatrix} I & J' \\ & J \end{bmatrix}^{\bowtie(d-2)} \bowtie \begin{bmatrix} J' \\ J \end{bmatrix}, \\ \boldsymbol{R}_{2^d}^{(d)} &= \begin{bmatrix} I & J \end{bmatrix} \bowtie \begin{bmatrix} I & J \\ & J' \end{bmatrix}^{\bowtie(d-2)} \bowtie \begin{bmatrix} J \\ J' \end{bmatrix}. \end{aligned}$$

Proof. The block structure

$$\boldsymbol{Q}_2^{(k)} = \begin{pmatrix} \boldsymbol{Q}_2^{(k-1)} & \\ J^{\otimes(k-1)} & \boldsymbol{Q}_2^{(k-1)} \end{pmatrix} = I \otimes \boldsymbol{Q}_2^{(k-1)} + J' \otimes J^{\otimes(k-1)},$$

of $\boldsymbol{Q}_2^{(k)}$ can be described in terms of the rank product as

$$\boldsymbol{Q}_2^{(k)} = \begin{bmatrix} I & J' \end{bmatrix} \bowtie \begin{bmatrix} \boldsymbol{Q}_2^{(k-1)} \\ J^{\otimes(k-1)} \end{bmatrix}, \tag{8}$$

which holds for $2 \le k \le d$. Let us apply (8) with $k = d-1$ to (8) with $k = d$:

$$\boldsymbol{Q}_2^{(d)} = \begin{bmatrix} I & J' \end{bmatrix} \bowtie \begin{bmatrix} I & J' & \\ & & J \end{bmatrix} \bowtie \begin{bmatrix} \boldsymbol{Q}_2^{(d-2)} \\ J^{\otimes(d-2)} \\ J^{\otimes(d-2)} \end{bmatrix}.$$

The latter decomposition is obviously redundant: we can exclude the third row of the right-hand core similarly to (7). We do this recursively and come to the decomposition to be proven:

$$\boldsymbol{Q}_2^{(d)} = \begin{bmatrix} I & J' \end{bmatrix} \bowtie \begin{bmatrix} I & J' \\ & J \end{bmatrix} \bowtie \begin{bmatrix} \boldsymbol{Q}_2^{(d-2)} \\ J^{\otimes(d-2)} \end{bmatrix} = \ldots = \begin{bmatrix} I & J' \end{bmatrix} \bowtie \begin{bmatrix} I & J' \\ & J \end{bmatrix}^{\bowtie(d-2)} \bowtie \begin{bmatrix} \boldsymbol{Q}_2^{(1)} \\ J \end{bmatrix}$$

$$= \begin{bmatrix} I & J' \end{bmatrix} \bowtie \begin{bmatrix} I & J' \\ & J \end{bmatrix}^{\bowtie(d-2)} \bowtie \begin{bmatrix} J' \\ J \end{bmatrix}.$$

The periodic shift matrix can be obtained as follows:

$$\boldsymbol{P}_2^{(d)} = \boldsymbol{Q}_2^{(d)} + \begin{pmatrix} & & 1 \end{pmatrix} = \boldsymbol{Q}_2^{(d)} + \begin{bmatrix} J \end{bmatrix}^{\bowtie d},$$

hence we can derive its decomposition from this sum by sweeping through the "tensor train" and excluding redundancy:

$$\begin{aligned} \boldsymbol{P}_2^{(d)} &= \begin{bmatrix} I & J' & J \end{bmatrix} \bowtie \begin{bmatrix} I & J' & \\ & J & \\ & & J \end{bmatrix}^{\bowtie(d-2)} \bowtie \begin{bmatrix} J' \\ J \\ J \end{bmatrix} \\ &= \begin{bmatrix} I & J' & J \end{bmatrix} \bowtie \begin{bmatrix} I & J' & \\ & J & \\ & & J \end{bmatrix}^{\bowtie(d-3)} \bowtie \begin{bmatrix} I & J' \\ & J \end{bmatrix} \bowtie \begin{bmatrix} J' \\ J \end{bmatrix} \\ &= \ldots = \begin{bmatrix} I & P \end{bmatrix} \bowtie \begin{bmatrix} I & J' \\ & J \end{bmatrix}^{\bowtie(d-2)} \bowtie \begin{bmatrix} J' \\ J \end{bmatrix}. \end{aligned}$$

As long as $\boldsymbol{R}_{2^d}^{(d)} = \boldsymbol{Q}_2^{(d)'}$, the decomposition of $\boldsymbol{R}_{2^d}^{(d)}$ follows trivially from the one of $\boldsymbol{Q}_2^{(d)'}$. □

Being applied to a vector of proper size, the two shift matrices considered above perform a one-position downward shift of its elements. We are interested in all shift matrices defining Toeplitz or, in particular, circulant matrix structure. So let us introduce also matrices

$$\boldsymbol{Q}_l^{(d)} = \left(\boldsymbol{Q}_2^{(d)}\right)^{l-1} = \begin{array}{c} \\ 1 \\ \vdots \\ l \\ \vdots \\ 2^d \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & \cdots & 2^d-l+1 & \cdots & 2^d \end{array} \\ \begin{pmatrix} & & & & \\ 1 & & & & \\ & \ddots & & & \\ & & 1 & & \end{pmatrix} \end{array},$$

$$\boldsymbol{R}_l^{(d)} = \left(\boldsymbol{R}_{2^d}^{(d)}\right)^{2^d-l+1} = \begin{array}{c} \\ 1 \\ \vdots \\ l-1 \\ \vdots \\ 2^d \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & \cdots & 2^d-l+2 & \cdots & 2^d \end{array} \\ \begin{pmatrix} & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ & & & & \end{pmatrix} \end{array},$$

and $\boldsymbol{P}_l^{(d)} = \left(\boldsymbol{P}_2^{(d)}\right)^{l-1} = \boldsymbol{Q}_l^{(d)} + \boldsymbol{R}_l^{(d)}$ for $1 \le l \le 2^d$ and also

$$\boldsymbol{S}_l^{(d)} = \begin{cases} \boldsymbol{R}_l^{(d)}, & 2 \le l \le 2^d, \\ \boldsymbol{Q}_{l-2^d}^{(d)}, & 2^d+1 \le l \le 2^{d+1}. \end{cases} \tag{9}$$

It is crucial for the course of the paper that, as we show below, these l-shift matrices can be represented in the QTT format with rank 2, altogether as well as singly.

Lemma 3.2. *Let $d \ge 2$ and $1 \le l \le 2^d$. Then the matrices $\boldsymbol{P}_l^{(k)}$, $\boldsymbol{Q}_l^{(k)}$, $\boldsymbol{R}_l^{(k)}$ and $\boldsymbol{S}_l^{(k)}$ have the following QTT representations:*

$$\begin{aligned} \boldsymbol{S}_{\overline{l_{d+1}\ldots l_1}}^{(d)} &= S_{l_{d+1}} \bowtie W_{l_d} \bowtie \ldots \bowtie W_{l_2} \bowtie V_{l_1}, \\ \boldsymbol{P}_{\overline{l_d\ldots l_1}}^{(d)} &= P_{l_d} \bowtie W_{l_{d-1}} \bowtie \ldots \bowtie W_{l_2} \bowtie V_{l_1}, \end{aligned}$$

$$\begin{aligned}\boldsymbol{Q}^{(d)}_{\overline{l_d \ldots l_1}} &= Q_{l_d} \bowtie W_{l_{d-1}} \bowtie \ldots \bowtie W_{l_2} \bowtie V_{l_1},\\ \boldsymbol{R}^{(d)}_{\overline{l_d \ldots l_1}} &= R_{l_d} \bowtie W_{l_{d-1}} \bowtie \ldots \bowtie W_{l_2} \bowtie V_{l_1},\end{aligned}$$

where $\overline{l_d \ldots l_1}$ *is a binary representation of* $l = 1 + \sum_{k=1}^{d} 2^{k-1}(l_k - 1)$ *and the cores are*

$$\begin{aligned} S_1 &= \begin{bmatrix}0 & 1\end{bmatrix}, & S_2 &= \begin{bmatrix}1 & 0\end{bmatrix},\\ P_1 &= \begin{bmatrix}I & P\end{bmatrix}, & P_2 &= \begin{bmatrix}P & I\end{bmatrix},\\ Q_1 &= \begin{bmatrix}I & J'\end{bmatrix}, & Q_2 &= \begin{bmatrix}J' & \end{bmatrix},\\ R_1 &= \begin{bmatrix} & J\end{bmatrix}, & R_2 &= \begin{bmatrix}J & I\end{bmatrix},\\ W_1 &= \begin{bmatrix}I & J'\\ & J\end{bmatrix}, & W_2 &= \begin{bmatrix}J' & \\ J & I\end{bmatrix},\\ V_1 &= \begin{bmatrix}I\end{bmatrix}, & V_2 &= \begin{bmatrix}J'\\ J\end{bmatrix}. \end{aligned} \tag{10}$$

Proof. For any $2 \le k \le d$ and $1 \le \lambda \le 2^{k-1}$ we have

$$\begin{aligned}\boldsymbol{Q}^{(k)}_{\lambda} &= I \otimes \boldsymbol{Q}^{(k-1)}_{\lambda} + J' \otimes \boldsymbol{R}^{(k-1)}_{\lambda},\\ \boldsymbol{R}^{(k)}_{\lambda} &= J \otimes \boldsymbol{R}^{(k-1)}_{\lambda},\\ \boldsymbol{Q}^{(k)}_{2^{k-1}+\lambda} &= J' \otimes \boldsymbol{Q}^{(k-1)}_{\lambda},\\ \boldsymbol{R}^{(k)}_{2^{k-1}+\lambda} &= J \otimes \boldsymbol{Q}^{(k-1)}_{\lambda} + I \otimes \boldsymbol{R}^{(k-1)}_{\lambda},\end{aligned}$$

which can be revised as

$$\begin{bmatrix}\boldsymbol{Q}^{(k)}_{\overline{l_k \ldots l_1}}\\ \boldsymbol{R}^{(k)}_{\overline{l_k \ldots l_1}}\end{bmatrix} = W_{l_k} \bowtie \begin{bmatrix}\boldsymbol{Q}^{(k-1)}_{\overline{l_{k-1} \ldots l_1}}\\ \boldsymbol{R}^{(k-1)}_{\overline{l_{k-1} \ldots l_1}}\end{bmatrix}$$

for $2 \le k \le d$. By applying the latter equation to itself recursively, we conclude that

$$\begin{bmatrix}\boldsymbol{Q}^{(d)}_{\overline{l_d \ldots l_1}}\\ \boldsymbol{R}^{(d)}_{\overline{l_d \ldots l_1}}\end{bmatrix} = W_{l_d} \bowtie W_{l_{d-1}} \bowtie \ldots W_{l_2} \bowtie \begin{bmatrix}\boldsymbol{Q}^{(1)}_{\overline{l_1}}\\ \boldsymbol{R}^{(1)}_{\overline{l_1}}\end{bmatrix} = \begin{bmatrix}Q_{l_d}\\ R_{l_d}\end{bmatrix} \bowtie W_{l_{d-1}} \bowtie \ldots W_{l_2} \bowtie V_{l_1},$$

which completes proof for $\boldsymbol{Q}^{(d)}_{\overline{l_d \ldots l_1}}$ and $\boldsymbol{R}^{(d)}_{\overline{l_d \ldots l_1}}$. Since $\boldsymbol{P}^{(d)}_{l} = \boldsymbol{Q}^{(d)}_{l} + \boldsymbol{R}^{(d)}_{l}$, $0 \le l \le 2^d - 1$, we come to the representation of $\boldsymbol{P}^{(d)}_{\overline{l_d \ldots l_1}}$:

$$\begin{aligned}\boldsymbol{P}^{(d)}_{\overline{l_d \ldots l_1}} &= \begin{bmatrix}1 & 1\end{bmatrix} \bowtie \begin{bmatrix}\boldsymbol{Q}^{(d)}_{\overline{l_d \ldots l_1}}\\ \boldsymbol{R}^{(d)}_{\overline{l_d \ldots l_1}}\end{bmatrix} = \begin{bmatrix}1 & 1\end{bmatrix} \bowtie W_{l_d} \bowtie W_{l_{d-1}} \bowtie \ldots W_{l_2} \bowtie V_{l_1}\\ &= P_{l_d} \bowtie W_{l_{d-1}} \bowtie \ldots W_{l_2} \bowtie V_{l_1}.\end{aligned}$$

The representation of $\boldsymbol{S}^{(d)}_{\overline{l_{d+1} \ldots l_1}}$ follows trivially from its definition (9) and the decompositions obtained just above. □

The decompositions elicited in Lemma 3.2 are remarkable owing to the fact that each core of them depends on the corresponding bit l_k of l only. This allows us to draw up at once a decomposition of the shift matrices as a whole. Let us stack the matrices $\boldsymbol{P}^{(d)}_{l}$, $1 \le l \le 2^d$, $\boldsymbol{Q}^{(d)}_{l}$, $1 \le l \le 2^d$, $\boldsymbol{R}^{(d)}_{l}$, $1 \le l \le 2^d$, and $\boldsymbol{S}^{(d)}_{l}$, $1 \le l \le 2^{d+1}$, into $2^d \times 2^d \times 2^d$-tensors $\boldsymbol{P}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ and a $2^d \times 2^d \times 2^{d+1}$-tensor $\boldsymbol{S}$ respectively so that

$$\boldsymbol{P}^{(d)}{}_{\cdot,\cdot,m} = \boldsymbol{P}^{(d)}_{m}(\cdot,\cdot), \quad \boldsymbol{Q}^{(d)}{}_{\cdot,\cdot,m} = \boldsymbol{Q}^{(d)}_{m}(\cdot,\cdot), \quad \boldsymbol{R}^{(d)}{}_{\cdot,\cdot,m} = \boldsymbol{R}^{(d)}_{m}(\cdot,\cdot)$$

for $1 \le m \le 2^d$ and $\boldsymbol{S}^{(d)}{}_{\cdot,\cdot,m} = \boldsymbol{S}^{(d)}_{m}(\cdot,\cdot)$ for $1 \le m \le 2^{d+1}$. Then QTT representations of these four tensors follow clearly from Lemma 3.2: we just recast its results by considering subscript indices of cores as their third mode indices.

Corollary 3.3. *Let $d \geq 2$. Then the tensors $\boldsymbol{S}^{(d)}$, $\boldsymbol{P}^{(d)}$, $\boldsymbol{Q}^{(d)}$ and $\boldsymbol{R}^{(d)}$ have the following rank-2 QTT representations:*

$$\begin{aligned}\boldsymbol{S}^{(d)} &= S \bowtie W^{\bowtie(d-1)} \bowtie V,\\ \boldsymbol{P}^{(d)} &= P \bowtie W^{\bowtie(d-2)} \bowtie V,\\ \boldsymbol{Q}^{(d)} &= Q \bowtie W^{\bowtie(d-2)} \bowtie V,\\ \boldsymbol{R}^{(d)} &= R \bowtie W^{\bowtie(d-2)} \bowtie V,\end{aligned}$$

where the TT cores are

$$S = \begin{bmatrix} 0|1 & 1|0 \end{bmatrix}, \quad P = \begin{bmatrix} I|P & P|I \end{bmatrix}, \quad Q = \begin{bmatrix} I|J' & J'|O \end{bmatrix}, \quad R = \begin{bmatrix} O|J & J|I \end{bmatrix},$$

$$W = \begin{bmatrix} I|J' & J'|O \\ O|J & J|I \end{bmatrix}, \quad V = \begin{bmatrix} I|J' \\ O|J \end{bmatrix}.$$

This modest result is a milestone in the analysis of QTT structure of multilevel Toeplitz matrices we do in this paper: the tensors $\boldsymbol{S}^{(d)}$, $\boldsymbol{P}^{(d)}$, $\boldsymbol{Q}^{(d)}$ and $\boldsymbol{R}^{(d)}$ define Toeplitz, circulant and lower and upper triangular Toeplitz structure at each level, respectively. We exploit this in the next section to decompose multilevel Toeplitz matrices in the QTT format.

3.2 Many dimensions

In the multidimensional case we deal with matrices of multidimensional shift of the form

$$\tilde{\boldsymbol{S}}_{l_D \ldots l_1} = \tilde{\boldsymbol{S}}^{(d_D)}_{l_D} \otimes \ldots \otimes \tilde{\boldsymbol{S}}^{(d_1)}_{l_1},$$

where each $\tilde{\boldsymbol{S}}^{(d_k)}_{l_k}$, $1 \leq k \leq D$, is a one-dimensional $l_k + 1$-shift $n_k \times n_k$-matrix; in particular, we may consider any of $\boldsymbol{S}^{(d_k)}_{l_k}$, $\boldsymbol{P}^{(d_k)}_{l_k}$, $\boldsymbol{Q}^{(d_k)}_{l_k}$ and $\boldsymbol{R}^{(d_k)}_{l_k}$ for each k. These matrices are stacked in a lexicographic order in the tensor

$$\tilde{\boldsymbol{S}} = \tilde{\boldsymbol{S}}^{(d_D)} \otimes \ldots \otimes \tilde{\boldsymbol{S}}^{(d_1)} = \tilde{\boldsymbol{S}}^{(d_D)} \bowtie \ldots \bowtie \tilde{\boldsymbol{S}}^{(d_1)},$$

where each tensor $\tilde{\boldsymbol{S}}^{(d_k)}$, $1 \leq k \leq D$, is $\boldsymbol{S}^{(d_k)}$, $\boldsymbol{P}^{(d_k)}$, $\boldsymbol{Q}^{(d_k)}$ or $\boldsymbol{R}^{(d_k)}$ correspondingly. Then $\tilde{\boldsymbol{S}}$ can be represented in the QTT format with ranks $2, \ldots, 2, \mathbf{1}, 2, \ldots, 2, \mathbf{1}, 2, \ldots, 2$ by Corollary 3.3. For example, a stack of two-dimensional downward periodic shift matrices can be written as

$$\boldsymbol{P} = \boldsymbol{P}^{(d_2)} \otimes \boldsymbol{P}^{(d_1)} = P \bowtie W^{\bowtie(d_2-2)} \bowtie V \bowtie P \bowtie W^{\bowtie(d_1-2)} \bowtie V,$$

where the cores P, W and V are the same as in Corollary 3.3. We use such multidimensional shift matrices to represent multilevel Toeplitz structure below.

4 Toeplitz structure in the QTT format

4.1 Structure of Toeplitz and circulant matrices

To start with, let us consider Toeplitz $n \times n$-matrices, where $n = 2^d$. Each of them is parameterized by $2n - 1$ entries of its first row and column. For any $2n$-component vector $\boldsymbol{x}$ we may consider a *Toeplitz matrix*

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{x}_{n+1} & \boldsymbol{x}_n & \cdots & \boldsymbol{x}_3 & \boldsymbol{x}_2 \\ \boldsymbol{x}_{n+2} & \ddots & \ddots & \ddots & \boldsymbol{x}_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{x}_{2n-1} & \ddots & \ddots & \ddots & \boldsymbol{x}_n \\ \boldsymbol{x}_{2n} & \boldsymbol{x}_{2n-1} & \cdots & \boldsymbol{x}_{n+2} & \boldsymbol{x}_{n+1} \end{pmatrix}, \tag{11}$$

which we refer to as a *Toeplitz matrix generated by the vector* $\boldsymbol{x}$. The component $\boldsymbol{x}_1$ is dummy and used for keeping the formal number of parameters even. In the particular cases

$$\boldsymbol{x} = \begin{pmatrix} \tilde{\boldsymbol{x}} \\ \tilde{\boldsymbol{x}} \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} 0 \\ \tilde{\boldsymbol{x}} \end{pmatrix}, \quad \text{and} \quad \boldsymbol{x} = \begin{pmatrix} \tilde{\boldsymbol{x}} \\ 0 \end{pmatrix},$$

where $\tilde{\boldsymbol{x}}$ is an n-component vector, we obtain a *circulant matrix*

$$\boldsymbol{C} = \begin{pmatrix} \tilde{\boldsymbol{x}}_1 & \tilde{\boldsymbol{x}}_n & \cdots & \tilde{\boldsymbol{x}}_3 & \tilde{\boldsymbol{x}}_2 \\ \tilde{\boldsymbol{x}}_2 & \ddots & \ddots & \ddots & \tilde{\boldsymbol{x}}_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \tilde{\boldsymbol{x}}_{n-1} & \ddots & \ddots & \ddots & \tilde{\boldsymbol{x}}_n \\ \tilde{\boldsymbol{x}}_n & \tilde{\boldsymbol{x}}_{n-1} & \cdots & \tilde{\boldsymbol{x}}_2 & \tilde{\boldsymbol{x}}_1 \end{pmatrix} \tag{12}$$

and *lower and upper triangular Toeplitz matrices*

$$\boldsymbol{L} = \begin{pmatrix} \tilde{\boldsymbol{x}}_1 & & & & \\ \tilde{\boldsymbol{x}}_2 & \ddots & & & \\ \vdots & \ddots & \ddots & & \\ \tilde{\boldsymbol{x}}_{n-1} & \ddots & \ddots & \ddots & \\ \tilde{\boldsymbol{x}}_n & \tilde{\boldsymbol{x}}_{n-1} & \cdots & \tilde{\boldsymbol{x}}_2 & \tilde{\boldsymbol{x}}_1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{U} = \begin{pmatrix} 0 & \tilde{\boldsymbol{x}}_n & \cdots & \tilde{\boldsymbol{x}}_3 & \tilde{\boldsymbol{x}}_2 \\ & \ddots & \ddots & \ddots & \tilde{\boldsymbol{x}}_3 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & \tilde{\boldsymbol{x}}_n \\ & & & & 0 \end{pmatrix} \tag{13}$$

generated by the vector $\tilde{\boldsymbol{x}}$.

The relation between a generating vector and the matrix generated by it in the case of the Toeplitz structure is often expressed in terms of shift matrices:

$$\boldsymbol{T} = \sum_{m=1}^{2^{d+1}} \boldsymbol{S}_m^{(d)} \boldsymbol{x}_m = \sum_{m=1}^{2^{d+1}} \boldsymbol{S}^{(d)}{}_{\cdot,\cdot,m} \boldsymbol{x}_m \equiv \boldsymbol{S}^{(d)} \bullet^3 \boldsymbol{x} \tag{14}$$

and by the same token

$$\boldsymbol{C} = \boldsymbol{P}^{(d)} \bullet^3 \tilde{\boldsymbol{x}}, \quad \boldsymbol{L} = \boldsymbol{Q}^{(d)} \bullet^3 \tilde{\boldsymbol{x}}, \quad \text{and} \quad \boldsymbol{U} = \boldsymbol{R}^{(d)} \bullet^3 \tilde{\boldsymbol{x}}. \tag{15}$$

The following four theorems describe QTT structure of matrices $\boldsymbol{T}$, $\boldsymbol{C}$, $\boldsymbol{L}$ and $\boldsymbol{U}$ generated by QTT-structured vectors $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$, as well as their products with a QTT-structured vector $\boldsymbol{y}$. Proof of all the four theorems follows at once from the Proposition 2.3, which we use to combine our result stated in Corollary 3.3 with the representations of Toeplitz and circulant matrices in terms of shift matrices (14), (15).

Theorem 4.1. *Let* $\boldsymbol{x}$ *be a* 2^{d+1}*-component vector,* $d \geq 2$*, given in a QTT representation*

$$\boldsymbol{x} = X_{d+1} \bowtie X_d \bowtie X_{d-1} \bowtie \ldots \bowtie X_2 \bowtie X_1$$

of ranks $p_d, p_{d-1}, \ldots, p_1$*. Then a Toeplitz* $2^d \times 2^d$*-matrix* $\boldsymbol{T}$ *generated by the vector* $\boldsymbol{x}$ *in the sense of* (11) *has a QTT decomposition*

$$\boldsymbol{T} = T_d \bowtie T_{d-1} \bowtie \ldots \bowtie T_2 \bowtie T_1$$

of ranks $2p_{d-1}, \ldots, 2p_1$*, composed of TT cores*

$$\begin{aligned} T_d &= (S \bullet X_{d+1}) \bowtie \left(W \bullet^3 X_d\right), \\ T_k &= W \bullet^3 X_k, \quad d-1 \geq k \geq 2, \\ T_1 &= V \bullet^3 X_1, \end{aligned}$$

where TT cores S*,* W *and* V *are the same as in Corollary 3.3.*

By relation (14), the matrix-vector product of a vector $\boldsymbol{y}$ and the Toeplitz matrix $\boldsymbol{T}$ generated by a vector $\boldsymbol{x}$ can be presented as

$$\boldsymbol{T}\boldsymbol{y} = \left(\boldsymbol{S}^{(d)} \bullet^3 \boldsymbol{x}\right) \cdot \boldsymbol{y} \equiv \left(\boldsymbol{S}^{(d)} \bullet^3 \boldsymbol{x}\right) \bullet^2 \boldsymbol{y} \equiv \boldsymbol{S}^{(d)} \bullet^{2,3} \boldsymbol{x}\boldsymbol{y}', \tag{16}$$

which leads us immediately to the following theorem.

Theorem 4.2. *Assume that $\boldsymbol{x}$ and $\boldsymbol{y}$ are 2^{d+1}-component and 2^d-component vectors respectively such that a $2^{d+1} \times 2^d$-matrix $\boldsymbol{x}\boldsymbol{y}'$ has a QTT representation*

$$\boldsymbol{x}\boldsymbol{y}' = G_{d+1} \bowtie G_d \bowtie G_{d-1} \bowtie \ldots \bowtie G_2 \bowtie G_1$$

of ranks $t_d, t_{d-1}, \ldots, t_1$. Then a matrix-vector product of a Toeplitz $2^d \times 2^d$-matrix $\boldsymbol{T}$ generated by the vector $\boldsymbol{x}$ in the sense of (11) *and the vector $\boldsymbol{y}$ has the following QTT decomposition of ranks $2t_{d-1}, \ldots, 2t_1$:*

$$\boldsymbol{T}\boldsymbol{y} = T_d \bowtie T_{d-1} \bowtie \ldots \bowtie T_2 \bowtie T_1,$$

where

$$\begin{aligned} T_d &= (S \bullet G_{d+1}) \bowtie \left(W \bullet^{2,3} G_d\right), \\ T_k &= W \bullet^{2,3} G_k, \quad d-1 \geq k \geq 2, \\ T_1 &= V \bullet^{2,3} G_1, \end{aligned}$$

the TT cores S, W and V being the same as in Corollary 3.3.

Remark 4.3. In Theorem 4.1 and Theorem 4.2 we come across QTT decompositions of tensors and matrices of inequal mode sizes, e. g. the tensor $\boldsymbol{S}$ of size $2^d \times 2^d \times 2^{d+1}$ and the matrix $\boldsymbol{x}\boldsymbol{y}'$ of size $2^{d+1} \times 2^d$. In these cases the highest (left-hand) cores in their QTT decompositions have fewer modes, e. g. either of S and G_{d+1} has only one mode index.

Similar theorems hold for circulant, lower triangular Toeplitz and upper triangular Toeplitz matrices.

Theorem 4.4. *Let $\tilde{\boldsymbol{x}}$ be a 2^d-component vector, $d \geq 2$, given in a QTT representation*

$$\tilde{\boldsymbol{x}} = \tilde{X}_d \bowtie \tilde{X}_{d-1} \bowtie \ldots \bowtie \tilde{X}_2 \bowtie \tilde{X}_1$$

of ranks $p_{d-1}, \ldots, p_1$. Then a circulant, lower triangular Toeplitz and upper triangular Toeplitz $2^d \times 2^d$-matrices $\boldsymbol{C}$, $\boldsymbol{L}$ and $\boldsymbol{U}$ generated by the vector $\tilde{\boldsymbol{x}}$ in the sense of (12) *and* (13) *have QTT decompositions*

$$\begin{aligned} \boldsymbol{C} &= \left(P \bullet^3 \tilde{X}_d\right) \bowtie \left(W \bullet^3 \tilde{X}_{d-1}\right) \bowtie \ldots \bowtie \left(W \bullet^3 \tilde{X}_2\right) \bowtie \left(V \bullet^3 \tilde{X}_1\right), \\ \boldsymbol{L} &= \left(Q \bullet^3 \tilde{X}_d\right) \bowtie \left(W \bullet^3 \tilde{X}_{d-1}\right) \bowtie \ldots \bowtie \left(W \bullet^3 \tilde{X}_2\right) \bowtie \left(V \bullet^3 \tilde{X}_1\right), \\ \boldsymbol{U} &= \left(R \bullet^3 \tilde{X}_d\right) \bowtie \left(W \bullet^3 \tilde{X}_{d-1}\right) \bowtie \ldots \bowtie \left(W \bullet^3 \tilde{X}_2\right) \bowtie \left(V \bullet^3 \tilde{X}_1\right) \end{aligned}$$

of ranks $2p_{d-1}, \ldots, 2p_1$, TT cores P, Q, R, W and V being the same as in Corollary 3.3.

Theorem 4.5. *Assume that $\tilde{\boldsymbol{x}}$ and $\boldsymbol{y}$ are 2^d-component vectors such that a $2^d \times 2^d$-matrix $\tilde{\boldsymbol{x}}\boldsymbol{y}'$ has a QTT representation*

$$\tilde{\boldsymbol{x}}\boldsymbol{y}' = \tilde{G}_{d+1} \bowtie \tilde{G}_d \bowtie \tilde{G}_{d-1} \bowtie \ldots \bowtie \tilde{G}_2 \bowtie \tilde{G}_1$$

of ranks $t_{d-1}, \ldots, t_1$. Then each of matrix-vector products of a circulant, lower triangular Toeplitz and upper triangular Toeplitz $2^d \times 2^d$-matrices $\boldsymbol{C}$, $\boldsymbol{L}$ and $\boldsymbol{U}$ generated by the vector $\tilde{\boldsymbol{x}}$ in the sense of (12), (13)*, and the vector $\boldsymbol{y}$ has a QTT representation $Z_d \bowtie Z_{d-1} \bowtie \ldots \bowtie Z_2 \bowtie Z_1$ of ranks $2t_{d-1}, \ldots, 2t_1$, where Z_d is equal to $P \bullet^{2,3} \tilde{G}_d$, $Q \bullet^{2,3} \tilde{G}_d$ or $R \bullet^{2,3} \tilde{G}_d$ in the cases of a circulant, lower triangular Toeplitz and upper triangular Toeplitz matrix respectively and $Z_k = W \bullet^{2,3} \tilde{G}_k$ for $d-1 \geq k \geq 2$, $Z_1 = V \bullet^{2,3} \tilde{G}_1$, where the TT cores P, Q, R, W and V are the same as in Corollary 3.3.*

4.2 Structure of a multilevel Toeplitz matrix

Let us now proceed to many dimensions. First, we put $N_1 = n_1$ and say that Toeplitz matrices $\boldsymbol{T}^{[1]}_{m_2}$, $1 \le m_2 \le 2n_2$, generated by $2n_1$-component vectors $\boldsymbol{x}^{[1]}_{m_2}$, are multilevel Toeplitz $N_1 \times N_1$-matrices with 1 level, generated by $2n_1$-tensors $\boldsymbol{x}^{[1]}_{m_2}$. Assume that $k \ge 1$ and multilevel Toeplitz $N_k \times N_k$-matrices $\boldsymbol{T}^{[k]}_{m_{k+1}}$, $1 \le m_{k+1} \le 2n_{k+1}$, with k levels, generated by $2n_1 \times \ldots \times 2n_k$-tensors $\boldsymbol{x}^{[k]}_{m_{k+1}}$, are defined. Let us put $N_{k+1} = n_{k+1} \cdot N_k$ and consider a Toeplitz-block $N_{k+1} \times N_{k+1}$-matrix

$$\boldsymbol{T}^{[k+1]} = \begin{pmatrix} \boldsymbol{T}^{[k]}_{n_{k+1}+1} & \boldsymbol{T}^{[k]}_{n_{k+1}} & \cdots & \boldsymbol{T}^{[k]}_{3} & \boldsymbol{T}^{[k]}_{2} \\ \boldsymbol{T}^{[k]}_{n_{k+1}+2} & \ddots & \ddots & \ddots & \boldsymbol{T}^{[k]}_{3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{T}^{[k]}_{2n_{k+1}-1} & \ddots & \ddots & \ddots & \boldsymbol{T}^{[k]}_{n_{k+1}} \\ \boldsymbol{T}^{[k]}_{2n_{k+1}} & \boldsymbol{T}^{[k]}_{2n_{k+1}-1} & \cdots & \boldsymbol{T}^{[k]}_{n_{k+1}+2} & \boldsymbol{T}^{[k]}_{n_{k+1}+1} \end{pmatrix} = \sum_{m_{k+1}=1}^{2n_{k+1}} \boldsymbol{S}^{(d_{k+1})}_{m_{k+1}} \otimes \boldsymbol{T}^{[k]}_{m_{k+1}} \tag{17}$$

as a multilevel Toeplitz $N_{k+1} \times N_{k+1}$-matrix with $k+1$ levels, generated by a $2n_1 \times \ldots \times 2n_{k+1}$-tensor $\boldsymbol{x}^{[k+1]}$ such that

$$\boldsymbol{x}^{[k+1]}{}_{m_1 \ldots m_{k+1}} = \left(\boldsymbol{x}^{[k]}_{m_{k+1}}\right)_{m_1 \ldots m_k}, \quad 1 \le m_\kappa \le 2n_\kappa, \quad 1 \le \kappa \le k+1.$$

A matrix $\boldsymbol{T} = \boldsymbol{T}^{[D]}$ defined by the recursion on k described above is called *multilevel Toeplitz with D levels*. We say that it is *generated by the tensor $\boldsymbol{x} = \boldsymbol{x}^{[D]}$*, which has D dimensions and mode size $2n_1 \times \ldots \times 2n_D$. This relation can also be expressed explicitly elementwise just in this way:

$$\boldsymbol{T}_{\boldsymbol{ij}} = \boldsymbol{x}_{\boldsymbol{i-j+n+1}}, \quad 1 \le i_k, j_k \le n_k, \quad 1 \le k \le D, \tag{18}$$

where $\boldsymbol{i} = (i_1, \ldots, i_D)$, $\boldsymbol{j} = (j_1, \ldots, j_D)$ and $\boldsymbol{n+1} = (n_1+1, \ldots, n_D+1)$ are multi-indices added and subtracted elementwise.

It might be easier to see from the recursive definition (17), that the multilevel Toeplitz matrix generated by a tensor may also be represented as a mode product of a "structuring tensor" and the generator:

$$\boldsymbol{T} = \boldsymbol{S} \bullet^3 \boldsymbol{x}, \tag{19}$$

where $\boldsymbol{S} = \boldsymbol{S}^{(d_D)} \otimes \ldots \otimes \boldsymbol{S}^{(d_1)}$. Then, like in the case of one-dimension, with the use of Proposition 2.3 we can apply the QTT representation of $\boldsymbol{S}$ following immediately from Corollary 3.3 and discussed briefly in Section 3.2.

Theorem 4.6. *Let $\boldsymbol{x}$ be a $2^{d_1+1} \times \ldots \times 2^{d_D+1}$-tensor, $D \ge 1$, given in a QTT representation*

$$\boldsymbol{x} = (X_{D,d_D+1} \bowtie \ldots \bowtie X_{D,1}) \bowtie \ldots, \ldots \bowtie (X_{1,d_1+1} \bowtie \ldots \bowtie X_{1,1})$$

of ranks $p_{D,d_D}, \ldots, p_{D,1}, \hat{p}_{D-1}, \ldots, \ldots, \hat{p}_1, p_{1,d_1}, \ldots, p_{1,1}$. Then a multilevel Toeplitz matrix $\boldsymbol{T}$ with D levels, generated by the tensor $\boldsymbol{x}$, has a QTT decomposition

$$\boldsymbol{T} = (T_{D,d_D} \bowtie \ldots \bowtie T_{D,1}) \bowtie \ldots, \ldots \bowtie (T_{1,d_1} \bowtie \ldots \bowtie T_{1,1})$$

of ranks $2p_{D,d_D-1}, \ldots, 2p_{D,1}, \hat{p}_{D-1}, \ldots, \ldots, \hat{p}_1, 2p_{1,d_1-1}, \ldots, 2p_{1,1}$, composed of the TT cores

$$\begin{aligned} T_{K,d_K} &= (S \bullet X_{K,d_K+1}) \bowtie \left(W \bullet^3 X_{K,d_K}\right), \\ T_{K,k} &= W \bullet^3 X_{K,k}, \quad d_K - 1 \ge k \ge 2, \\ T_{K,1} &= V \bullet^3 X_{K,1}, \end{aligned}$$

$1 \le K \le D$, where the TT cores S, W and V are the same as in Corollary 3.3.

Similarly to (14), the matrix-vector product of a vector $\boldsymbol{y}$ and the multilevel Toeplitz matrix $\boldsymbol{T}$ generated by a vector $\boldsymbol{x}$ can be presented as

$$\boldsymbol{Ty} = \left(\boldsymbol{S} \bullet^3 \boldsymbol{x}\right) \cdot \boldsymbol{y} \equiv \left(\boldsymbol{S} \bullet^3 \boldsymbol{x}\right) \bullet^2 \boldsymbol{y} \equiv \boldsymbol{S} \bullet^{2,3} \boldsymbol{xy}'. \tag{20}$$

This yields immediately the following result.

Theorem 4.7. *Assume that $\boldsymbol{x}$ and $\boldsymbol{y}$ are $2^{d_1+1} \times \ldots \times 2^{d_D+1}$ and $2^{d_1} \times \ldots \times 2^{d_D}$-tensors respectively such that a $\left(2^{d_1+1} \times \ldots \times 2^{d_D+1}\right) \times \left(2^{d_1} \times \ldots \times 2^{d_D}\right)$-matrix $\boldsymbol{x}\boldsymbol{y}'$ has a QTT representation*

$$\boldsymbol{x}\boldsymbol{y}' = (G_{D,d_D+1} \bowtie \ldots \bowtie G_{D,1}) \bowtie \ldots, \ldots \bowtie (G_{1,d_1+1} \bowtie \ldots \bowtie G_{1,1})$$

of ranks $t_{D,d_D}, \ldots, t_{D,1}, \hat{t}_{D-1}, \ldots, \ldots, \hat{t}_1, t_{1,d_1}, \ldots, t_{1,1}$. Then a matrix-vector product of a multilevel Toeplitz matrix $\boldsymbol{T}$ with D levels, generated by the tensor $\boldsymbol{x}$, and the tensor $\boldsymbol{y}$ has the following QTT decomposition of ranks $2t_{D,d_D-1}, \ldots, 2t_{D,1}, \hat{t}_{D-1}, \ldots, \ldots, \hat{t}_1, 2t_{1,d_1-1}, \ldots, 2t_{1,1}$:

$$\boldsymbol{T}\boldsymbol{y} = (Z_{D,d_D} \bowtie \ldots \bowtie Z_{D,1}) \bowtie \ldots, \ldots \bowtie (Z_{1,d_1} \bowtie \ldots \bowtie Z_{1,1}),$$

where

$$\begin{aligned} Z_{K,d_K} &= (S \bullet G_{K,d_K+1}) \bowtie \left(W \bullet^{2,3} G_{K,d_K}\right), \\ Z_{K,k} &= W \bullet^{2,3} G_{K,k}, \quad d_K - 1 \geq k \geq 2, \\ Z_{K,1} &= V \bullet^{2,3} G_{K,1}, \end{aligned}$$

$1 \leq K \leq D$, the TT cores S, W and V being the same as in Corollary 3.3.

The multilevel matrix structure defined recursively by (17) and explicitly by (18) encompasses circulant, lower and upper triangular structure as well. For example, we may impose a requirement $\boldsymbol{x}_{\boldsymbol{m}} = \boldsymbol{x}_{\boldsymbol{m}+n_k\boldsymbol{e}_{\boldsymbol{k}}}$ for $1 \leq m_\kappa \leq 2n_\kappa$, $\kappa \neq k$, and $n_k + 1 \leq m_k \leq 2n_k$, where the multi-index $\boldsymbol{e}_{\boldsymbol{k}} = (0, \ldots 0, 1, 0, \ldots 0)$ has 1 at the k-th position, on the generator $\boldsymbol{x}$, then the matrix $\boldsymbol{T}$ can be said to have *circulant structure at the k-th level*, and similarly for the triangular Toeplitz structures. In this cases we may reduce the k-th mode size of the generator $\boldsymbol{x}$ from $2n_k$ down to n_k. Then Theorem 4.6 and Theorem 4.7 can be also specified to exclude redundant computations.

Remark 4.8. Once QTT representations of a 2^{d+1}-component vector $\boldsymbol{x}$ and a 2^d-component vector $\boldsymbol{y}$ of ranks $p_d, p_{d-1}, \ldots p_1$ and $q_{d-1}, \ldots q_1$ respectively are given, the matrix $\boldsymbol{x}\boldsymbol{y}' \equiv \boldsymbol{x} \bullet_1^2 \boldsymbol{y}'$ can be trivially decomposed in the QTT format with ranks $p_d, p_{d-1}q_{d-1}, \ldots p_1q_1$. This gives us an upper bound of the QTT ranks of the vector $\boldsymbol{T}\boldsymbol{y}$ itself: they are not higher than $2p_d, 2p_{d-1}q_{d-1}, \ldots 2p_1q_1$, as follows from Theorem 4.2, and this upper bound appears to be sharp in numerical experiments, which is a reliable evidence since the TT arithmetics is robust. Also, this suggests us a naive way of computing the matrix-vector product with a Toeplitz matrix. However, ranks of the decomposition of $\boldsymbol{x}\boldsymbol{y}'$ assumed to be given in Theorem 4.2 may be remarkably lower than $p_d, p_{d-1}q_{d-1}, \ldots p_1q_1$ and can be found out by standard matrix algorithms (see Introductions and references therein).

4.3 Discrete convolution in the QTT format

We also study discrete convolution, which is closely related to multilevel Toeplitz matrices, as we point out in this section. We start with convolution $\boldsymbol{h} = \boldsymbol{f} \star \boldsymbol{g}$ of functions $\boldsymbol{f}$ and $\boldsymbol{g}$ of a $\mathbb{Z}$-valued argument, that is

$$\boldsymbol{h}_i = \sum_{j=-\infty}^{+\infty} \boldsymbol{f}_{i-j}\boldsymbol{g}_j, \quad i \in \mathbb{Z}. \tag{21}$$

One of or both the functions convolved are periodic. If $\boldsymbol{f}$ or $\boldsymbol{g}$ is periodic with a period n, then (21) can be recast as

$$\boldsymbol{h}_i = \sum_{j=0}^{n-1} \boldsymbol{f}_{i-j}\overline{\boldsymbol{g}}_j = \sum_{j=0}^{n-1} \boldsymbol{f}_{(i-j) \bmod n}\overline{\boldsymbol{g}}_j \tag{22}$$

$$\text{or} \quad \boldsymbol{h}_i = \sum_{j=0}^{n-1} \overline{\boldsymbol{f}}_{i-j}\boldsymbol{g}_j = \sum_{j=0}^{n-1} \overline{\boldsymbol{f}}_{(i-j) \bmod n}\boldsymbol{g}_j \tag{23}$$

respectively, where $\overline{\boldsymbol{g}}_j = \sum_{s=-\infty}^{+\infty} \boldsymbol{g}_{j+sn}$ and $\overline{\boldsymbol{f}}_j = \sum_{s=-\infty}^{+\infty} \boldsymbol{f}_{j-sn}$ are *periodic summations* of $\boldsymbol{g}$ and $\boldsymbol{f}$. Both the operands are now periodic with the period n.

When both the functions to be convolved are periodic; for example, if they are considered on a contour, (21) makes no sense in a nontrivial case. Alternatively, the following is usually meant by convolution of two periodic functions of a $\mathbb{Z}$-valued argument with a common period n:

$$\boldsymbol{h}_i = \sum_{j=0}^{n-1} \boldsymbol{f}_{i-j}\boldsymbol{g}_j = \sum_{j=0}^{n-1} \boldsymbol{f}_{(i-j) \bmod n}\boldsymbol{g}_j, \quad i \in \mathbb{Z}. \tag{24}$$

The convolutions (22), (23), (24) discussed above have the same form

$$\boldsymbol{h}_i = \sum_{j=0}^{n-1} \hat{\boldsymbol{f}}_{(i-j) \bmod n}\hat{\boldsymbol{g}}_j, \quad i \in \mathbb{Z}, \tag{25}$$

which is defined for $i = 0, \ldots, n-1$ as a vector $\boldsymbol{z} = \begin{pmatrix}\boldsymbol{h}_0 & \cdots & \boldsymbol{h}_{n-1}\end{pmatrix}'$ by vectors $\boldsymbol{x} = \begin{pmatrix}\hat{\boldsymbol{f}}_0 & \cdots & \hat{\boldsymbol{f}}_{n-1}\end{pmatrix}'$ and $\boldsymbol{y} = \begin{pmatrix}\hat{\boldsymbol{g}}_0 & \cdots & \hat{\boldsymbol{g}}_{n-1}\end{pmatrix}'$ and to be continued periodically to $\mathbb{Z}$. In this regard we can recast (25) with the help of the circulant matrix structure (12) as follows:

$$\boldsymbol{z} = \boldsymbol{C} \cdot \boldsymbol{y}, \tag{26}$$

where $\boldsymbol{C}$ is a circulant matrix generated by the vector $\boldsymbol{x}$.

Both the functions convolved have compact supports. In this case we assume that $\boldsymbol{f}_j = \boldsymbol{g}_j = 0$, $j \neq 0, \ldots, n-1$ for some $n \in \mathbb{N}$ in (21). Then the convolution $\boldsymbol{h}$ of $\boldsymbol{f}$ and $\boldsymbol{g}$ is nonzero at $2n-1$ points only and equals

$$\boldsymbol{h}_i = \begin{cases} \sum_{j=0}^{n-1} \boldsymbol{f}_{i-j}\boldsymbol{g}_j, & 0 \leq i \leq 2n-2, \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

It is defined for $i = 0, \ldots, 2n-1$ as a vector $\boldsymbol{z} = \begin{pmatrix}\boldsymbol{h}_0 & \cdots & \boldsymbol{h}_{2n-2} & 0\end{pmatrix}'$ by vectors $\boldsymbol{x} = \begin{pmatrix}\boldsymbol{f}_0 & \cdots & \boldsymbol{f}_{n-1}\end{pmatrix}'$ and $\boldsymbol{y} = \begin{pmatrix}\boldsymbol{g}_0 & \cdots & \boldsymbol{g}_{n-1}\end{pmatrix}'$ and can be expressed in terms of matrix-vector multiplication as

$$\boldsymbol{z} = \tilde{\boldsymbol{L}} \cdot \boldsymbol{y} = \boldsymbol{T} \cdot \begin{pmatrix}\boldsymbol{y} \\ 0\end{pmatrix}, \tag{28}$$

where

$$\tilde{\boldsymbol{L}} = \begin{pmatrix} \boldsymbol{x}_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \boldsymbol{x}_n & \ddots & \ddots & \boldsymbol{x}_1 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \boldsymbol{x}_n \\ 0 & \cdots & \cdots & 0 \end{pmatrix} \tag{29}$$

and $\boldsymbol{T}$ is a Toeplitz matrix generated by the vector $\begin{pmatrix}0 & 0 & \boldsymbol{x}' & 0\end{pmatrix}'$.

One of the functions convolved has compact support. If we assume only that $\boldsymbol{g}_j = 0$, $j \neq 0, \ldots, n-1$ for some $n \in \mathbb{N}$ in (21), i. e. $\boldsymbol{g}$ has compact support while $\boldsymbol{f}$ may not, their convolution $\boldsymbol{h}$ cannot be expected to have compact support either. But values of $\boldsymbol{f}$ at the points $1, \ldots, 2n-1$ and all the non-zero values of $\boldsymbol{g}$ allow us to calculate

$$\boldsymbol{h}_i = \sum_{j=0}^{n-1} \boldsymbol{f}_{i-j}\boldsymbol{g}_j, \quad n \leq i \leq 2n-1. \tag{30}$$

Let us compose vectors $\boldsymbol{x} = \begin{pmatrix} * & \boldsymbol{f}_1 & \cdots & \boldsymbol{f}_{2n-1} \end{pmatrix}'$ and $\boldsymbol{y} = \begin{pmatrix} \boldsymbol{g}_0 & \cdots & \boldsymbol{g}_{n-1} \end{pmatrix}'$, then (30) can be rewritten for a vector $\boldsymbol{z} = \begin{pmatrix} \boldsymbol{h}_n & \cdots & \boldsymbol{h}_{2n-1} \end{pmatrix}'$ as

$$\boldsymbol{z} = \boldsymbol{T} \cdot \boldsymbol{y}, \tag{31}$$

the matrix $\boldsymbol{T}$ being Toeplitz generated by the vector $\boldsymbol{x}$.

Discrete convolution in many dimensions and Toeplitz matrix structure. In D dimensions we deal with convolution $\boldsymbol{h} = \boldsymbol{f} \star \boldsymbol{g}$ of functions $\boldsymbol{f}$ and $\boldsymbol{g}$ of a $\mathbb{Z}^D$-valued argument, which generalizes (21) and (24) to the following:

$$\boldsymbol{h_i} = \sum_{\boldsymbol{j}} \boldsymbol{f_{i-j}} \boldsymbol{g_j}, \tag{32}$$

where $\boldsymbol{i} = (i_1, \ldots, i_D)$ and $\boldsymbol{j} = (j_1, \ldots, j_D)$ are multi-indices, which are added and subtracted elementwise. For each k the range of i_k and summation limits of j_k depend on the particular kind of convolution applied with respect to the k-th dimension. We emphasized it above that in one dimension in each of the particular cases we have considered so far convolution can be obtained as a product of a matrix and a vector composed of values of $\boldsymbol{g}$, the matrix being Toeplitz generated by a vector of values of $\boldsymbol{f}$. Therefore, a vector $\boldsymbol{z}$ of values of the convolution $\boldsymbol{h}$ (32), on the condition that it is considered in the sense of (25), (27) or (30) with respect to each level, can be calculated from tensors $\boldsymbol{x}$ and $\boldsymbol{y}$ of values of $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively as

$$\boldsymbol{z} = \boldsymbol{T} \cdot \boldsymbol{y}, \tag{33}$$

where $\boldsymbol{T}$ is a multilevel Toeplitz matrix with D levels, generated by the tensor $\boldsymbol{x}$.

5 How to compute multilevel Toeplitz matrices and their products with vectors in the QTT format

In Section 4.2 we applied the QTT decomposition of the "structuring" tensor $\boldsymbol{S}$, following from Corollary 3.3, to relations (19) and (20) in order to describe QTT structure of a multilevel Toeplitz matrix $\boldsymbol{T}$ and its product $\boldsymbol{Ty}$ with a multidimensional vector. Though Theorem 4.6 and Theorem 4.7 present these results in terms of single TT cores, the theorems actually propose nothing else than matrix-vector multiplication in the QTT format for evaluation of $\boldsymbol{T}$ and $\boldsymbol{Ty}$.

The most straightforward approach to compute $\boldsymbol{T}$ is to exploit the QTT structure of matrix-vector multiplication, which allows to rewrite it core-wise (see Proposition 2.3) explicitly and perform it exactly in this very way. We assume that the generator $\boldsymbol{x}$ is given in a QTT representation of reasonable ranks: if it is not the case, we can truncate it in a robust way and get rid of overestimated ranks (see references on TT in Introduction). As long as construction of a Toeplitz matrix from a vector boosts QTT ranks not more than by the factor of 2, we may carry it out by means of the exact matrix-vector multiplication, as we describe in Algorithm 5.1, and wind up with still feasible ranks.

However, the rank issue gets severe if we compute a matrix-vector product of $\boldsymbol{T}$ and another vector $\boldsymbol{y}$. Exact computation according to equation (20) (or Theorem 4.7) by two successive tensor contractions, which are actually matrix-vector multiplications of properly reshaped data in this case, leads to a QTT representation of the result with QTT ranks up to $2pq$, where p and q are the corresponding ranks of $\boldsymbol{x}$ and $\boldsymbol{y}$. Complexity of TT arithmetics being polynomial w. r. t. QTT ranks, such high ranks hinder further computations with the output, while the actual QTT structure of it might be not worse or even better than that of the input vectors (which can be observed as a smoothening effect of convolution in the case of function-related data). To obtain a truncated decomposition of the result, which is the same tractable in terms of QTT rank structure as the input QTT representations of $\boldsymbol{x}$ and $\boldsymbol{y}$, several approaches may be brought into play.

The simplest idea is to employ the innate QR-SVD-based TT truncation of the output decomposition of multiplied ranks, which can be done by $\mathcal{O}\left(Dd \cdot 2 \cdot (2pq)^3\right) = \mathcal{O}\left(Dd\, p^3 q^3\right)$ operations with a

Algorithm 5.1 $\boldsymbol{T} = \texttt{tt\_qtoepl}(\boldsymbol{x})$

Require: QTT decomposition $\boldsymbol{S} = S \bowtie W^{\bowtie(d_D-1)} \bowtie V \bowtie \ldots, \ldots \bowtie S \bowtie W^{\bowtie(d_1-1)} \bowtie V$; exact TT matrix-vector multiplication subroutine `tt_mv`

Input: a $2^{d_1+1} \times \ldots \times 2^{d_D+1}$-tensor $\boldsymbol{x}$ in a QTT decomposition $\boldsymbol{x} = (X_{D,d_D+1} \bowtie \ldots \bowtie X_{D,1}) \bowtie \ldots, \ldots \bowtie (X_{1,d_1+1} \bowtie \ldots \bowtie X_{1,1})$ of ranks $p_{D,d_D}, \ldots, p_{D,1}$, $\hat{p}_{D-1}, \ldots, \ldots, \hat{p}_1$, $p_{1,d_1}, \ldots, p_{1,1}$

Output: the multilevel Toeplitz matrix $\boldsymbol{T}$ generated by the tensor $\boldsymbol{x}$ in a QTT decomposition $\boldsymbol{T} = (T_{D,d_D} \bowtie \ldots \bowtie T_{D,1}) \bowtie \ldots, \ldots \bowtie (T_{1,d_1} \bowtie \ldots \bowtie T_{1,1})$ of ranks $2p_{D,d_D-1}, \ldots, 2p_{D,1}$, $\hat{p}_{D-1}, \ldots, \ldots, \hat{p}_1$, $2p_{1,d_1-1}, \ldots, 2p_{1,1}$

1: merge indices i and j in cores V and W {reshape $\boldsymbol{S}$ into a matrix of size $4^{\sum_{k=1}^{D} d_k} \times 2^{\sum_{k=1}^{D}(d_k+1)}$}

2: $\boldsymbol{T} = \texttt{tt\_mv}(\boldsymbol{S}, \boldsymbol{x})$ {exact matrix-vector multiplication in the TT format}

3: split indices i and j in cores $T_{K,k}$, $d_K \geq k \geq 1$, $D \geq K \geq 1$ {reshape $\boldsymbol{T}$ into a matrix of size $2^{\sum_{k=1}^{D} d_k} \times 2^{\sum_{k=1}^{D} d_k}$}

D-dimensional $2^d \times \ldots \times 2^d$-vector of rank $2pq$. However, the cubic dependence of the complexity on either of the ranks makes such an approach unfavorable for even moderate p and q.

Another and much more efficient way is to perform the most expensive tensor contraction, which involves both the input vectors, approximately rather than exactly. This allows to accomplish the truncation on the fly and attain a proper rank structure of the output without constructing its exact representation of QTT rank $2pq$.

Algorithm 5.2 relying on an abstract matrix-vector multiplication subroutine fulfills both the approaches, depending on what particular subroutine is used. It may be either exact or inexact and require some extra arguments (e. g. accuracy), which should be also passed to the algorithm. Subsequent truncation, if needed (for example, in the case of exact matrix-vector multiplication), should be applied additionally afterwards.

Algorithm 5.2 $\boldsymbol{z} = \texttt{tt\_qconv\_x}(\boldsymbol{x}, \boldsymbol{y}, \ldots)$

Require: multilevel Toeplitz matrix construction subroutine `tt_qtoepl`; TT matrix-vector multiplication subroutine `tt_mv_x`

Input: a $2^{d_1+1} \times \ldots \times 2^{d_D+1}$-tensor $\boldsymbol{x}$ and a $2^{d_1} \times \ldots \times 2^{d_D}$-tensor $\boldsymbol{y}$ in QTT decompositions
$\boldsymbol{x} = (X_{D,d_D+1} \bowtie \ldots \bowtie X_{D,1}) \bowtie \ldots, \ldots \bowtie (X_{1,d_1+1} \bowtie \ldots \bowtie X_{1,1})$ and
$\boldsymbol{y} = (Y_{D,d_D} \bowtie \ldots \bowtie Y_{D,1}) \bowtie \ldots, \ldots \bowtie (Y_{1,d_1} \bowtie \ldots \bowtie Y_{1,1})$ of ranks
$p_{D,d_D}, \ldots, p_{D,1}$, $\hat{p}_{D-1}, \ldots, \ldots, \hat{p}_1$, $p_{1,d_1}, \ldots, p_{1,1}$ and
$q_{D,d_D-1}, \ldots, q_{D,1}$, $\hat{q}_{D-1}, \ldots, \ldots, \hat{q}_1$, $q_{1,d_1-1}, \ldots, q_{1,1}$ respectively

Output: matrix-vector product of $\boldsymbol{y}$ and the multilevel Toeplitz matrix generated by $\boldsymbol{x}$ in a QTT decomposition $\boldsymbol{z} = (Z_{D,d_D} \bowtie \ldots \bowtie Z_{D,1}) \bowtie \ldots, \ldots \bowtie (Z_{1,d_1} \bowtie \ldots \bowtie Z_{1,1})$ of the ranks bounded from above by $2p_{D,d_D-1}q_{D,d_D-1}, \ldots, 2p_{D,1}q_{D,1}$, $\hat{p}_{D-1}\hat{q}_{D-1}, \ldots, \ldots, \hat{p}_1\hat{q}_1$, $2p_{1,d_1-1}q_{1,d_1-1}, \ldots, 2p_{1,1}q_{1,1}$

1: $\boldsymbol{T} = \texttt{tt\_qtoepl}(\boldsymbol{x})$ {construction of the multilevel Toeplitz matrix generated by $\boldsymbol{x}$}

2: $\boldsymbol{z} = \texttt{tt\_mv\_x}(\boldsymbol{T}, \boldsymbol{y}, \ldots)$ {matrix-vector multiplication in the TT format}

We propose to use the iterative DMRG matrix-vector multiplication procedure described in [42] to compute $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{y} = \boldsymbol{x} \star \boldsymbol{y}$ with a prescribed relative ℓ_2-accuracy ε. In order to estimate clearly the complexity of such a procedure and Algorithm 5.2 based on it, let us assume that $d_k = d$ for $k = 1, \ldots, D$, p and q bound from above QTT ranks of $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively, r bounds from above those of the ε-approximation $\boldsymbol{z}_\varepsilon$ of $\boldsymbol{z}$ being computed. Then the complexity of a single iteration of the DMRG matrix-vector multiplication for $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{y}$ reads

$$\begin{aligned}
&\mathcal{O}\left(Dd\left[2 \cdot r^3 + 2^2 \cdot (2p)^2 \cdot q \cdot r + 2^2 \cdot 2p \cdot q^2 \cdot r + 2 \cdot 2p \cdot q \cdot r^2\right]\right) \\
&\quad = \mathcal{O}\left(Dd\, r^3 + Dd\, pqr\,(p + q + r)\right),
\end{aligned}$$

which agrees to the complexity estimate in [42, Section 3.2] if we assume additionally that $r = q$, and

exact construction of $\boldsymbol{T}$ in the QTT format according to Algorithm 5.1 still costs $\mathcal{O}\left(Dd\,p^2\right)$, which is negligible.

6 Numerical experiments

We present the following numerical experiments of convolution of vectors in one and three dimensions with the use of Algorithm 5.2 implemented in MATLAB. A workhorse of our computations is the TT Toolbox developed by Ivan Oseledets with contributions from his colleagues at the Institute of Numerical Mathematics of Russian Academy of Sciences and publicly available at `http://spring.inm.ras.ru/osel`. We use the following functions of the toolbox:

- $(\boldsymbol{T}, \boldsymbol{y}) \mapsto \boldsymbol{z} = \texttt{tt\_mv}(\boldsymbol{T}, \boldsymbol{y})$ multiplies a matrix $\boldsymbol{T}$ by a vector $\boldsymbol{y}$ in the QTT format exactly corewise as described in Proposition 2.3;
- $\boldsymbol{z} \mapsto \boldsymbol{z}_\varepsilon = \texttt{tt\_compr2}(\boldsymbol{z}, \varepsilon)$ truncates a TT representation of $\boldsymbol{z}$ passed in a quasi-optimal way with a given relative accuracy ε in the Frobenius norm;
- $(\boldsymbol{T}, \boldsymbol{y}, \varepsilon) \mapsto \boldsymbol{z}_\varepsilon = \texttt{tt\_mv\_k2}(\boldsymbol{T}, \boldsymbol{y}, \varepsilon)$ multiplies a matrix $\boldsymbol{T}$ by a vector $\boldsymbol{y}$ in the TT format with the use of the DMRG approach with a given relative accuracy ε in the Frobenius norm;

and also `full_to_tt` to approximate tensors in the TT format, `tt_dist2` to compute the Frobenius distance between two given tensors and `tt_dot` to compute their dot product. Details on truncation and exact matrix-vector multiplication in the TT format can be found in the papers on the TT format referred to in Introduction. The DMRG algorithm for matrix-vector multiplication is presented in [42].

We propose the following three implementations of Algorithm 5.2, discussed in Section 5:

- **"exact"**: $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \boldsymbol{z} = \texttt{tt\_qconv}(\boldsymbol{x}, \boldsymbol{y})$ computes the convolution exactly ($\texttt{tt\_mv\_x} = \texttt{tt\_mv}$);
- **"exact + tr."**: $(\boldsymbol{x}, \boldsymbol{y}, \varepsilon) \mapsto \boldsymbol{z}_\varepsilon = \texttt{tt\_compr2}(\texttt{tt\_qconv}(\boldsymbol{x}, \boldsymbol{y}), \varepsilon)$ computes the convolution exactly and truncates it with a prescribed accuracy ε ($\texttt{tt\_mv\_x} = \texttt{tt\_mv}$);
- **"DMRG"**: $(\boldsymbol{x}, \boldsymbol{y}, \varepsilon) \mapsto \boldsymbol{z}_\varepsilon = \texttt{tt\_qconv\_k2}(\boldsymbol{x}, \boldsymbol{y})$ computes the convolution approximately with a prescribed accuracy ε with the use of the DMRG matrix-vector multiplication ($\texttt{tt\_mv\_x} = \texttt{tt\_mv\_k2}$).

In order to present concisely rank structure of the QTT decompositions involved in our numerical experiments, we utilize the following two widely used aggregate rank characteristics. Let us consider a $n_1 \times \ldots n_d$-tensor given in a TT decomposition, the rank structure of which is described completely by $d-1$ ranks $r_1, \ldots, r_{d-1}$. Then we refer to $r_{\max} = \max_{k=1,\ldots,d-1} r_k$ as the *maximum rank* of the decomposition in question, while its *effective rank* r_{eff} is defined by the equation

$$n_1 r_1 + \sum_{k=2}^{d-1} r_{k-1} n_k r_k + r_{d-1} n_d, = n_1 r_{\text{eff}} + \sum_{k=2}^{d-1} r_{\text{eff}} n_k r_{\text{eff}} + r_{\text{eff}} n_d$$

which equals memory needed to store the decomposition given and a decomposition of ranks $r_{\text{eff}}, \ldots, r_{\text{eff}}$ of the tensor given, so that the "effective rank" is meant to be effective w. r. t. memory. But it allows to evaluate exactly complexity of some operations in the TT format, such as matrix-vector multiplication and Hadamard product, and gives a reasonable measure of complexity of others, e. g. TT truncation.

All the numerical experiments presented below were executed in MATLAB 2009b on a single core of a CPU Intel Xeon E5504 2.00GHz with 72 Gb memory available. We compare our results to the preprint [9], experiments for which were carried out on the same computer and on a single core as well, but in FORTRAN. The latter circumstance should be taken into account, as long as MATLAB and FORTRAN implementations of TT truncation and matrix-vector multiplication might differ in

performance more remarkably than those of the Fast Fourier Transform calling external more or less similarly optimized subroutines.

Note that we do not compare our algorithm to the CP-structured convolution based on one-dimensional FFT directly (see (5) with explanation in Section 1.1 or [23, 20] for more details), as long as such a comparison follows trivially from collation of our algorithm and the FFT-based convolution in the full format presented in Section 6.1.

6.1 Convolution of random QTT-structured vectors in 1D

To start with, we consider periodic convolution of vectors with random QTT decompositions of prescribed ranks in one dimension and compare our approach to the FFT-based convolution in the full format. For a given dimensionality d and rank r, we generate QTT cores of 2^d-vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ as arrays of random numbers uniformly distributed in $[0, 1]$. All the ranks of the cores are let equal to r, so both the effective and maximum ranks of the decompositions also equal r.

In the QTT format ranks are bounded by minimal sizes of the corresponding unfolding matrices (see Introduction and references therein), which are $2^1, 2^2, 2^3, \ldots, 2^3, 2^2, 2^1$. Therefore, ranks of the decompositions generated in the way described above may be reduced by exact transformations of cores similar to (7) down to $\min\left(r, 2^1\right)$, $\min\left(r, 2^2\right)$, $\min\left(r, 2^3\right), \ldots, \min\left(r, 2^3\right)$, $\min\left(r, 2^2\right)$, $\min\left(r, 2^1\right)$, which are actual QTT ranks of the random tensors involved. However, in this experiment we use rank-$r, \ldots, r$ decompositions to make it easier to track how the performance of convolution depends on $r = r_{\text{eff}} = r_{\max}$.

Regarding the choice of the one-dimensional case of 2^d-vectors for this experiment, we point out that for random vectors it cannot differ any significantly from the D-dimensional case of $2^{d_1} \times \ldots \times 2^{d_D}$-vectors once $d = d_1 + \ldots + d_D$, as long as there is no special "interaction" between dimensions in a random tensor and our convolution algorithm is even a little faster in higher dimensions due to the fact that the QTT ranks of the structuring tensor $\boldsymbol{S}$, connecting "real" dimensions, are equal to 1 instead of 2.

Time of periodic convolution is presented in Table 1 and Figure 1 for $r = 5$, 15, 40. Truncation parameter ε is equal to 10^{-2} in both the approximate implementations of Algorithm 5.2, which ensures the accuracy 10^{-2} of the result.

We compare the algorithm proposed to the FFT-based convolution algorithm consisting of three FFTs and a single Hadamard product in the full format. Note that we do note take into account any expenses related to conversion from the full format to QTT or vice versa, we just assume that convolution is done in either of the two and compare corresponding times. Generally speaking, we are interested in convolution in time logarithmic w. r. t. the number of components of the input vectors, which could be efficient as a stage of a particular computational process carried out in the QTT format, while QTT approximation of full vectors and their convolution with subsequent conversion back to the full format do not make any sense and are not available in more high-dimensional problems, which we deal with in other numerical experiments. Of course, performance of the FFT-based algorithm is in no way affected by QTT structure (e. g. ranks) of the random vectors in question.

We also consider the times presented in [9] multiplied by the factor of three in order to make a comparison to the convolution algorithm based on QTT-FFT. In that paper computation of FFT of a sum of r complex planes with random amplitudes and frequencies was considered, so the input decompositions had ranks $r, \ldots, r$ and the vectors themselves were of ranks $\min\left(r, 2^1\right)$, $\min\left(r, 2^2\right)$, $\min\left(r, 2^3\right), \ldots, \min\left(r, 2^3\right)$, $\min\left(r, 2^2\right)$, $\min\left(r, 2^1\right)$, which conforms to our experiment. We assume that introduction of the factor of three leads to a reasonable estimate of QTT-FFT convolution time, but, rigorously speaking, the difference in QTT structure of the data may lead to that in performance. Because of this we postpone a rigorous comparison of our methods to the QTT-FFT-based convolution till Section 6.2.2. One may look at the times of FFT-based convolution we measured and those of FFT presented in [9] and tripled by us to compare performance of these computations in MATLAB and FORTRAN and take the corresponding scaling into account while comparing the two QTT-structured convolution algorithms. For them, however, the difference between MATLAB and FORTRAN might

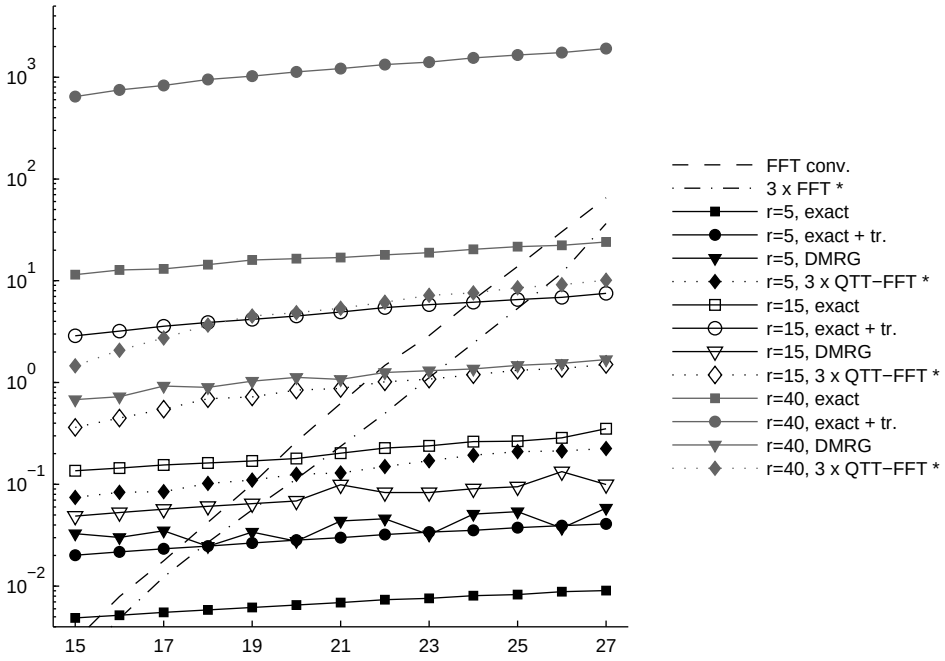


Figure 1: Time (sec.) of convolution of random 2^d-vectors vs. d. The four lines marked with * present time of three FFTs and QTT-FFTs of a sum of r plane waves with random amplitudes and frequencies.

be more noticeable than for the standard FFT subroutines.

The experiment shows that Algorithm 5.2 outperforms the convolution method based on FFT in the full format when the vectors are long enough and their QTT ranks are reasonably bounded from above. The crossing points starting from which QTT convolution performs better are $d = 16$ for $r = 5$ (exact version), $d = 18$ for $r = 15$ (DMRG version) and $d = 22$ for $r = 40$ (DMRG version). For practical ranks (say, more than 10) exact convolution becomes too expensive compared to the DMRG version and subsequent truncation of the output to reasonable ranks takes a lot of extra time. Also, for all the three values of r considered the DMRG version is faster than we can expect the QTT-FFT convolution method to be, while the exact version and that with subsequent truncation excel it only for moderate ranks. So we may suggest the DMRG version of Algorithm 5.2 as the method of choice.

6.2 Convolution of function-related vectors in 3D

As we mentioned in Section 1.1, convolution plays important role in scientific computing and, in particular, computational quantum chemistry and solid state physics. Gaussian Type Orbitals (GTO), which are Gaussians with polynomial weights [43], were and are still widely used to represent electronic and molecular structure, potentials and all the intermediate data in plenty of commercial and non-commercial software packages, e. g. MOLPRO, GAMESS, Q-Chem, CASINO, ABINIT, etc. A great advantage of GTOs is that they are highly capable of representing strong cusps typical for this kind of applications and suitable for analytical calculations.

Alternatively, tensor methods (in the Tucker [44, 45], CP, TT or QTT format) may be applied directly to electronic density functions, potentials and other data and operators involved in computations (see [22, 23, 25, 24] for details and further references). This propels us to consider convolution of a Gaussian with another Gaussian (Section 6.2.1) and the Newton kernel (Section 6.2.2) as examples in the case of function-related data. Convolution $f \star g \in L_r\left(\mathbb{R}^D\right)$ of functions $f \in L_p\left(\mathbb{R}^D\right)$ and $g \in L_q\left(\mathbb{R}^D\right)$ is defined as $(f \star g)(u) = \int_{\mathbb{R}^D} f(u-v)g(v)\mathrm{d}v$, $u \in \mathbb{R}^D$, and by *Young's inequality for convolutions* we have for $1 \le p, q, r \le \infty$ that

$$\|f \star g\|_{L_r(\mathbb{R}^D)} \le \|f\|_{L_p(\mathbb{R}^D)} \|g\|_{L_q(\mathbb{R}^D)}, \quad \text{provided that} \quad 1 + \frac{1}{r} = \frac{1}{p} + \frac{1}{q}, \tag{34}$$

	in the full format		in the QTT format				
d	FFT conv.	3×FFT *	r	exact	exact+tr.	DMRG	3×QTT-FFT *
15	$2.9 \cdot 10^{-3}$	$1.8 \cdot 10^{-3}$	5	$4.9 \cdot 10^{-3}$	$2.0 \cdot 10^{-2}$	$3.3 \cdot 10^{-2}$	$7.4 \cdot 10^{-2}$
			15	$1.4 \cdot 10^{-1}$	$2.9 \cdot 10^{+0}$	$4.9 \cdot 10^{-2}$	$3.6 \cdot 10^{-1}$
			40	$1.1 \cdot 10^{+1}$	$6.4 \cdot 10^{+2}$	$6.8 \cdot 10^{-1}$	$1.5 \cdot 10^{+0}$
16	$7.9 \cdot 10^{-3}$	$4.8 \cdot 10^{-3}$	5	$5.2 \cdot 10^{-3}$	$2.2 \cdot 10^{-2}$	$3.0 \cdot 10^{-2}$	$8.3 \cdot 10^{-2}$
			15	$1.4 \cdot 10^{-1}$	$3.2 \cdot 10^{+0}$	$5.3 \cdot 10^{-2}$	$4.5 \cdot 10^{-1}$
			40	$1.3 \cdot 10^{+1}$	$7.5 \cdot 10^{+2}$	$7.3 \cdot 10^{-1}$	$2.1 \cdot 10^{+0}$
17	$1.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	5	$5.5 \cdot 10^{-3}$	$2.3 \cdot 10^{-2}$	$3.5 \cdot 10^{-2}$	$8.5 \cdot 10^{-2}$
			15	$1.6 \cdot 10^{-1}$	$3.6 \cdot 10^{+0}$	$5.7 \cdot 10^{-2}$	$5.5 \cdot 10^{-1}$
			40	$1.3 \cdot 10^{+1}$	$8.3 \cdot 10^{+2}$	$9.2 \cdot 10^{-1}$	$2.7 \cdot 10^{+0}$
18	$4.2 \cdot 10^{-2}$	$2.7 \cdot 10^{-2}$	5	$5.8 \cdot 10^{-3}$	$2.5 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$
			15	$1.6 \cdot 10^{-1}$	$3.9 \cdot 10^{+0}$	$6.1 \cdot 10^{-2}$	$6.9 \cdot 10^{-1}$
			40	$1.4 \cdot 10^{+1}$	$9.5 \cdot 10^{+2}$	$9.0 \cdot 10^{-1}$	$3.7 \cdot 10^{+0}$
19	$1.0 \cdot 10^{-1}$	$5.6 \cdot 10^{-2}$	5	$6.2 \cdot 10^{-3}$	$2.7 \cdot 10^{-2}$	$3.4 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$
			15	$1.7 \cdot 10^{-1}$	$4.2 \cdot 10^{+0}$	$6.5 \cdot 10^{-2}$	$7.2 \cdot 10^{-1}$
			40	$1.6 \cdot 10^{+1}$	$1.0 \cdot 10^{+3}$	$1.0 \cdot 10^{+0}$	$4.5 \cdot 10^{+0}$
20	$2.6 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	5	$6.5 \cdot 10^{-3}$	$2.8 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$
			15	$1.8 \cdot 10^{-1}$	$4.5 \cdot 10^{+0}$	$6.9 \cdot 10^{-2}$	$8.4 \cdot 10^{-1}$
			40	$1.7 \cdot 10^{+1}$	$1.1 \cdot 10^{+3}$	$1.1 \cdot 10^{+0}$	$4.8 \cdot 10^{+0}$
21	$6.3 \cdot 10^{-1}$	$2.4 \cdot 10^{-1}$	5	$6.9 \cdot 10^{-3}$	$3.0 \cdot 10^{-2}$	$4.4 \cdot 10^{-2}$	$1.3 \cdot 10^{-1}$
			15	$2.0 \cdot 10^{-1}$	$4.9 \cdot 10^{+0}$	$9.9 \cdot 10^{-2}$	$8.7 \cdot 10^{-1}$
			40	$1.7 \cdot 10^{+1}$	$1.2 \cdot 10^{+3}$	$1.1 \cdot 10^{+0}$	$5.4 \cdot 10^{+0}$
22	$1.5 \cdot 10^{+0}$	$5.0 \cdot 10^{-1}$	5	$7.4 \cdot 10^{-3}$	$3.2 \cdot 10^{-2}$	$4.6 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$
			15	$2.3 \cdot 10^{-1}$	$5.5 \cdot 10^{+0}$	$8.3 \cdot 10^{-2}$	$1.0 \cdot 10^{+0}$
			40	$1.8 \cdot 10^{+1}$	$1.3 \cdot 10^{+3}$	$1.3 \cdot 10^{+0}$	$6.1 \cdot 10^{+0}$
23	$2.9 \cdot 10^{+0}$	$1.1 \cdot 10^{+0}$	5	$7.6 \cdot 10^{-3}$	$3.4 \cdot 10^{-2}$	$3.2 \cdot 10^{-2}$	$1.7 \cdot 10^{-1}$
			15	$2.4 \cdot 10^{-1}$	$5.8 \cdot 10^{+0}$	$8.3 \cdot 10^{-2}$	$1.1 \cdot 10^{+0}$
			40	$1.9 \cdot 10^{+1}$	$1.4 \cdot 10^{+3}$	$1.3 \cdot 10^{+0}$	$7.2 \cdot 10^{+0}$
24	$6.5 \cdot 10^{+0}$	$2.4 \cdot 10^{+0}$	5	$8.1 \cdot 10^{-3}$	$3.5 \cdot 10^{-2}$	$5.1 \cdot 10^{-2}$	$1.9 \cdot 10^{-1}$
			15	$2.6 \cdot 10^{-1}$	$6.2 \cdot 10^{+0}$	$9.0 \cdot 10^{-2}$	$1.2 \cdot 10^{+0}$
			40	$2.0 \cdot 10^{+1}$	$1.5 \cdot 10^{+3}$	$1.4 \cdot 10^{+0}$	$7.6 \cdot 10^{+0}$
25	$1.4 \cdot 10^{+1}$	$5.4 \cdot 10^{+0}$	5	$8.3 \cdot 10^{-3}$	$3.8 \cdot 10^{-2}$	$5.4 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$
			15	$2.7 \cdot 10^{-1}$	$6.5 \cdot 10^{+0}$	$9.5 \cdot 10^{-2}$	$1.3 \cdot 10^{+0}$
			40	$2.2 \cdot 10^{+1}$	$1.7 \cdot 10^{+3}$	$1.5 \cdot 10^{+0}$	$8.5 \cdot 10^{+0}$
26	$3.0 \cdot 10^{+1}$	$1.2 \cdot 10^{+1}$	5	$8.8 \cdot 10^{-3}$	$3.9 \cdot 10^{-2}$	$3.7 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$
			15	$2.9 \cdot 10^{-1}$	$6.9 \cdot 10^{+0}$	$1.3 \cdot 10^{-1}$	$1.4 \cdot 10^{+0}$
			40	$2.2 \cdot 10^{+1}$	$1.7 \cdot 10^{+3}$	$1.5 \cdot 10^{+0}$	$9.2 \cdot 10^{+0}$
27	$6.6 \cdot 10^{+1}$	$3.6 \cdot 10^{+1}$	5	$9.0 \cdot 10^{-3}$	$4.1 \cdot 10^{-2}$	$5.9 \cdot 10^{-2}$	$2.3 \cdot 10^{-1}$
			15	$3.5 \cdot 10^{-1}$	$7.5 \cdot 10^{+0}$	$1.0 \cdot 10^{-1}$	$1.5 \cdot 10^{+0}$
			40	$2.4 \cdot 10^{+1}$	$1.9 \cdot 10^{+3}$	$1.7 \cdot 10^{+0}$	$1.0 \cdot 10^{+1}$

Table 1: Time (sec.) of convolution of random 2^d-vectors. The two columns marked with * present time of three FFTs and QTT-FFTs of a sum of r plane waves with random amplitudes and frequencies.

which is very useful for us in view of the convolution error control, since convolution is a bilinear mapping.

In the following experiments we consider $h = f \star g$ in three dimensions ($D = 3$) with a Gaussian g defined by

$$g(u) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^3} \exp\left(-\frac{\|u\|^2}{2\sigma^2}\right), \quad u \in \mathbb{R}^3,$$

where we let $\sigma = 10^{-3}$, so that g represents a strong cusp at 0, which requires very careful interpolation to be resolved. The value of σ also allows us to approximate g very accurately by a finite function

$$\tilde{g}(u) = \begin{cases} g(u), & u \in \left[-\frac{1}{2}, \frac{1}{2}\right]^3, \\ 0, & \text{otherwise}, \end{cases}$$

which is feasible by (34). The function f is considered on $[-1,1]$, and $f \star \tilde{g}$ is defined then on $\left[-\frac{1}{2}, \frac{1}{2}\right]$. We assume that both the functions convolved are centered in a way, which is not restrictive due to the *translation property of convolution*.

To proceed from continuous convolution to (30) and (31), we use a piecewise-constant FEM discretization of f on $[-1,1]^3$ and g on $\left[-\frac{1}{2}, \frac{1}{2}\right]^3$, and a piecewise-multilinear FEM discretization of h on $\left[-\frac{1}{2}, \frac{1}{2}\right]^3$. Finite elements are constructed on the tensor grids $\left(\left\{\frac{2i-1-2n}{2n}\right\}_{i=1}^{2n}\right)^{\times 3}$, $\left(\left\{\frac{2i-1-n}{2n}\right\}_{i=1}^{n}\right)^{\times 3}$ and $\left(\left\{\frac{2i-n}{2n}\right\}_{i=1}^{n}\right)^{\times 3}$ respectively, where $n = 2^d$.

Next we approximate the FEM discretizations $\boldsymbol{x}$, $\boldsymbol{y}$ and $\hat{\boldsymbol{z}}$ of f, g and h respectively in the QTT format, compute $\boldsymbol{z} = \boldsymbol{x} \star \boldsymbol{y}$ by the three versions of Algorithm 5.2 described at the beginning of Section 6 and examine the error $\boldsymbol{e} = \boldsymbol{z} - \hat{\boldsymbol{z}}$. The latter is done in two ways: we rely mostly on the relative ℓ_2-norm $\delta = \frac{\|\boldsymbol{e}\|_2}{\|\hat{\boldsymbol{z}}\|_2}$ of the error, which we evaluate in the QTT format. But, following [9], we also *est*imate accuracy by the relative ℓ_2-norm δ_{est} of $\boldsymbol{e}$ restricted to the axes. Computation of δ_{est}, which reduces to one dimension, is easier and affordable in the full format with no need to approximate $\hat{\boldsymbol{z}}$ in the QTT format. We also keep an eye on the relative ℓ_1 and Chebyshev norms of $\boldsymbol{e}$ restricted to the axes, which prove to follow the same tendencies as δ_{est} in our examples and, thus, are not reported for the sake of brevity.

To obtain QTT approximations $\boldsymbol{y}$, we take advantage of the perfect separability of a Gaussian: it is a tensor power of the proper one-dimensional Gaussian $g^{(1)}$ which we represent in the corresponding finite element subspace, reshape the discretization into a $2 \times \ldots \times 2$-tensor and approximate with in the TT format by the TT Toolbox subroutine `full_to_tt` in order to obtain its QTT approximation $\boldsymbol{y}^{(1)}$. We end up with $\boldsymbol{y} = \boldsymbol{y}^{(1)} \bowtie \boldsymbol{y}^{(1)} \bowtie \boldsymbol{y}^{(1)}$, which has small QTT ranks and approximates the discretization of g with relative accuracies $\lesssim 10^{-13}$ in the ℓ_2, ℓ_1 and Chebyshev norms, while $r_{\text{eff}} \leq 6$ and $r_{\max} \leq 12$. Since in our experiments f and h are either Gaussians as well or can be approximated by sums of those, we use the same approach to obtain $\boldsymbol{x}$ and $\hat{\boldsymbol{z}}$ in the QTT format.

Remark 6.1. The error analysis presented in [20] applies to computation of convolution with the discretization described above. In particular, the estimate $\|\boldsymbol{z} - \hat{\boldsymbol{z}}\|_{\text{C}} = \mathcal{O}\left(\frac{1}{n^2}\right)$, $n \to \infty$ [20, Theorem 2.2] holds for $\boldsymbol{z}$ if all the QTT approximations involved are absolutely accurate.

6.2.1 Convolution of two Gaussians in 3D

In this experiment we let f be a Gaussian:

$$f(u) = \frac{1}{\left(\sqrt{2\pi}\sigma_0\right)^3} \exp\left(-\frac{\|u\|^2}{2\sigma_0^2}\right), \quad u \in [-1,1]^3$$

with $\sigma_0 = 1$. A simple calculation shows that the convolution result h is another Gaussian:

$$h(u) = \frac{1}{\left(\sqrt{2\pi\left(\sigma_0^2 + \sigma^2\right)}\right)^3} \exp\left(-\frac{\|u\|^2}{2\left(\sigma_0^2 + \sigma^2\right)}\right), \quad u \in \left[-\frac{1}{2}, \frac{1}{2}\right]^3.$$

As one can see in Figure 2(a), Figure 3(a), Figure 4(a) and Table 2, the accurate and low-rank QTT approximations of discretizations, which are available in the case of Gaussians, unfortunately, do not allow us to track the asymptotics of the convolution accuracy with respect to d, predicted by Remark 6.1: relative Frobenius-norm convolution error is too small with respect to the norm of the result when the FEM approximation is still not very accurate (small d).

d	ε	$r^{\boldsymbol{z}}_{\text{eff}}$	$r^{\boldsymbol{z}}_{\max}$	δ	δ_{est}	time
	exact					
10		17.0	24	$6.2 \cdot 10^{-9}$	$6.2 \cdot 10^{-9}$	**0.009**
11		22.2	32	$1.1 \cdot 10^{-14}$	$2.2 \cdot 10^{-14}$	**0.010**
12		26.9	64	$1.2 \cdot 10^{-14}$	$1.7 \cdot 10^{-14}$	**0.011**
14		38.6	96	$3.3 \cdot 10^{-14}$	$4.1 \cdot 10^{-14}$	**0.013**
16		42.3	96	$3.1 \cdot 10^{-14}$	$1.4 \cdot 10^{-14}$	**0.017**
18		43.2	96	$7.5 \cdot 10^{-14}$	$6.6 \cdot 10^{-14}$	**0.018**
20		51.3	120	$1.9 \cdot 10^{-13}$	$6.1 \cdot 10^{-14}$	**0.022**
	exact + truncation					
10	$5.0 \cdot 10^{-9}$	3.2	4	$6.2 \cdot 10^{-9}$	$7.1 \cdot 10^{-9}$	**0.013**
11	$5.0 \cdot 10^{-14}$	4.2	6	$1.4 \cdot 10^{-14}$	$1.2 \cdot 10^{-14}$	**0.018**
12	$5.0 \cdot 10^{-14}$	4.2	6	$1.3 \cdot 10^{-14}$	$7.6 \cdot 10^{-15}$	**0.022**
14	$5.0 \cdot 10^{-14}$	4.1	6	$2.8 \cdot 10^{-14}$	$2.3 \cdot 10^{-14}$	**0.042**
16	$5.0 \cdot 10^{-14}$	3.9	6	$3.2 \cdot 10^{-14}$	$3.4 \cdot 10^{-14}$	**0.061**
18	$5.0 \cdot 10^{-14}$	3.8	6	$8.2 \cdot 10^{-14}$	$5.0 \cdot 10^{-14}$	**0.067**
20	$5.0 \cdot 10^{-14}$	3.8	6	$1.8 \cdot 10^{-13}$	$7.9 \cdot 10^{-14}$	**0.108**
	DMRG					
10	$5.0 \cdot 10^{-9}$	5.0	6	$6.6 \cdot 10^{-9}$	$6.6 \cdot 10^{-9}$	**0.084**
11	$5.0 \cdot 10^{-13}$	5.7	7	$4.3 \cdot 10^{-13}$	$9.3 \cdot 10^{-13}$	**0.096**
12	$5.0 \cdot 10^{-13}$	5.6	7	$6.4 \cdot 10^{-13}$	$1.2 \cdot 10^{-12}$	**0.106**
14	$5.0 \cdot 10^{-13}$	5.6	7	$8.1 \cdot 10^{-13}$	$1.5 \cdot 10^{-12}$	**0.128**
16	$5.0 \cdot 10^{-13}$	5.5	7	$7.8 \cdot 10^{-13}$	$1.5 \cdot 10^{-12}$	**0.152**
18	$5.0 \cdot 10^{-13}$	5.4	7	$8.1 \cdot 10^{-13}$	$1.2 \cdot 10^{-12}$	**0.166**
20	$5.0 \cdot 10^{-13}$	5.3	7	$8.3 \cdot 10^{-13}$	$1.3 \cdot 10^{-12}$	**0.187**

Table 2: Convolution of two Gaussians. Time is given in seconds

The convolution graphs are similar to the case of random low-rank ($r = 5$) vectors in one dimension (Section 6.1): the DMRG version of Algorithm 5.2 is slower than the exact one with subsequent truncation. However, the results presented in Section 6.1 suggest that this changes if we deal with higher QTT ranks, which is typical for more practical examples.

6.2.2 Newton potential of a Gaussian in 3D

Another example is the Newton potential of the Gaussian g, which is the result h of convolution of g with the Newton kernel f defined by

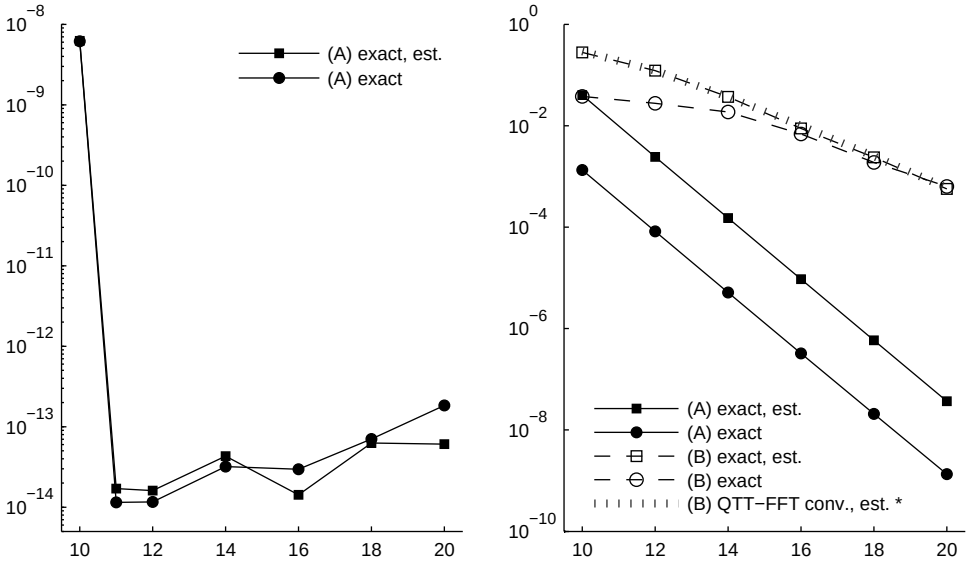
$$f(u) = \frac{1}{\|u\|}, \quad u \in [-1, 1]^3.$$

The convolution result h is expressed analytically in terms of the error function as follows [43, pp. 806–813]:

$$h(u) = \frac{1}{\|u\|} \operatorname{erf}\left(\frac{\|u\|}{\sqrt{2}\sigma}\right), \quad u \in \left[-\frac{1}{2}, \frac{1}{2}\right]^3$$

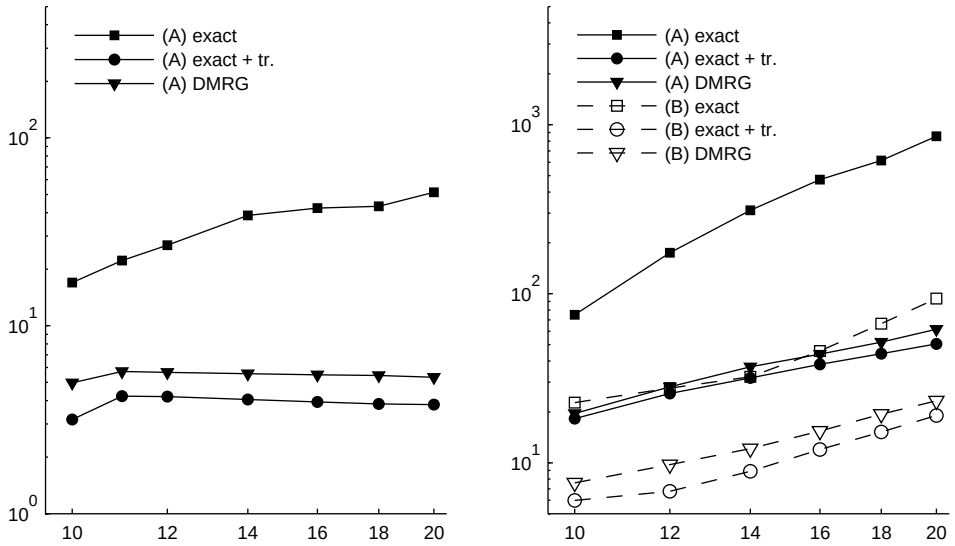
The discretization applied to f (see Section 6.2) allows us to disregard the singularity at 0 in rather a naive way, which, however, still yields us a good approximation of the convolution in the end, as we mentioned in Remark 6.1.

We take advantage of the quadratures presented in [46], which represent the functions $r^2 \mapsto \frac{1}{r}$ and $r^2 \mapsto \frac{1}{r}\operatorname{erf}\left(\frac{1}{r}\right)$, $r > 0$, as sums of exponentials, to decompose f and h in sums of Gaussians. Then we discretize and approximate each Gaussian in the QTT format in the same way as g, sum them and compress the results with $\mathrm{tt\_compr2}$. This yields us $\boldsymbol{x}$ and $\hat{\boldsymbol{z}}$ in the QTT format. We estimate the accuracy of the decompositions obtained on the axes similarly to how δ_{est} is computed (more



(a) Convolution of two Gaussians (b) Newton potential of a Gaussian

Figure 2: Accuracy of the convolution vs. d



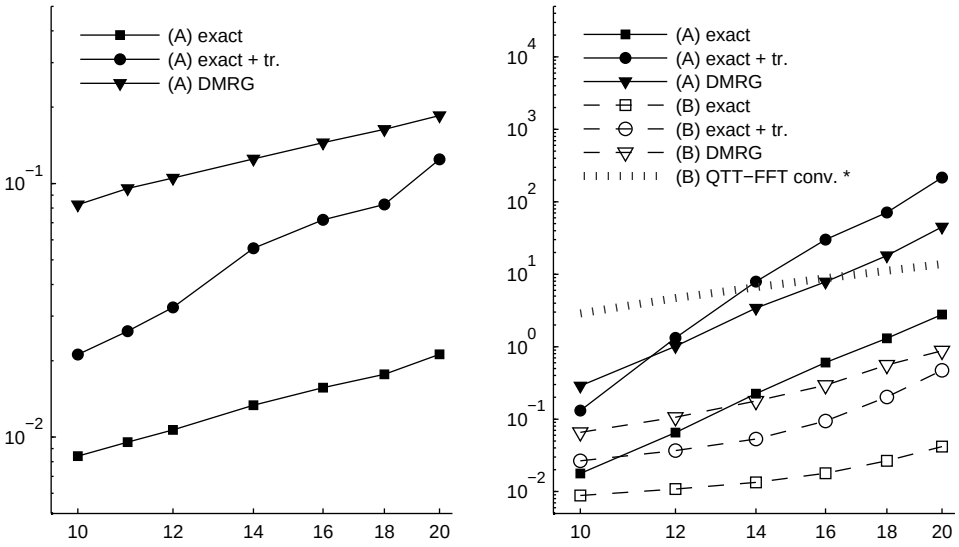
(a) Convolution of two Gaussians (b) Newton potential of a Gaussian

Figure 3: Effective rank of the convolution vs. d

precisely, the accuracy of $\boldsymbol{x}$ is estimated at the distance $\frac{1}{2n}$ from each of the axis due to our choice of discretization of f). The relative accuracy estimate proves to be $\lesssim 10^{-11}$ in the ℓ_2, ℓ_1 and Chebyshev norms.

As we will see from the experiment, the decompositions of $\boldsymbol{x}$ and $\boldsymbol{y}$ obtained as described above are a bit too accurate and can be truncated before we apply Algorithm 5.2. This is important for $\boldsymbol{x}$, which has rather high ranks; for example, $r_{\text{eff}}^{\boldsymbol{x}} = 308$ and $r_{\max}^{\boldsymbol{x}} \approx 174$ for $d = 20$ in our experiments. We call `tt_compr2` with a truncation parameter $\varepsilon^{\boldsymbol{x}} = \varepsilon^{\boldsymbol{y}}$ to approximate $\boldsymbol{x}$ and $\boldsymbol{y}$ with smaller QTT ranks. This gives rise to two series of experiments, which we denote by "(A)" and "(B)". The truncation parameters applied and the characteristic ranks obtained are given in Table 3 and Table 4.

For the series (A) the input data truncation parameter $\varepsilon^{\boldsymbol{x}} = \varepsilon^{\boldsymbol{y}}$ is chosen so that the accuracy δ of the output is about the best possible. For the series (B) we choose such $\varepsilon^{\boldsymbol{x}} = \varepsilon^{\boldsymbol{y}}$ that the accuracy of the output is about the same as and not worse than that reported by the authors of [9] (the results included in the current version of the preprint [9] are not the best: we found experiments with better



(a) Convolution of two Gaussians (b) Newton potential of a Gaussian

Figure 4: Convolution time (sec.) vs. d

	approximation of f			approximation of g		
d	$\varepsilon^{\boldsymbol{x}}$	$r^{\boldsymbol{x}}_{\text{eff}}$	$r^{\boldsymbol{x}}_{\max}$	$\varepsilon^{\boldsymbol{y}}$	$r^{\boldsymbol{y}}_{\text{eff}}$	$r^{\boldsymbol{y}}_{\max}$
10	$2.0 \cdot 10^{-4}$	19.6	31	$2.0 \cdot 10^{-4}$	1.9	2
12	$1.0 \cdot 10^{-5}$	31.3	50	$1.0 \cdot 10^{-5}$	2.8	6
14	$1.0 \cdot 10^{-6}$	43.2	70	$1.0 \cdot 10^{-6}$	3.6	7
16	$1.0 \cdot 10^{-7}$	56.9	94	$1.0 \cdot 10^{-7}$	4.1	8
18	$1.0 \cdot 10^{-8}$	72.5	120	$1.0 \cdot 10^{-8}$	4.2	8
20	$1.0 \cdot 10^{-9}$	91.4	152	$1.0 \cdot 10^{-9}$	4.6	9

Table 3: Newton potential of a Gaussian (A). Ranks and accuracy of the input vectors

	approximation of f			approximation of g		
d	$\varepsilon^{\boldsymbol{x}}$	$r^{\boldsymbol{x}}_{\text{eff}}$	$r^{\boldsymbol{x}}_{\max}$	$\varepsilon^{\boldsymbol{y}}$	$r^{\boldsymbol{y}}_{\text{eff}}$	$r^{\boldsymbol{y}}_{\max}$
10	$5.3 \cdot 10^{-2}$	6.1	9	$5.3 \cdot 10^{-2}$	1.9	2
12	$4.3 \cdot 10^{-2}$	6.7	10	$4.3 \cdot 10^{-2}$	2.2	4
14	$3.2 \cdot 10^{-2}$	7.6	12	$3.2 \cdot 10^{-2}$	2.3	4
16	$1.2 \cdot 10^{-2}$	10.3	17	$1.2 \cdot 10^{-2}$	2.3	4
18	$3.6 \cdot 10^{-3}$	14.7	26	$3.6 \cdot 10^{-3}$	2.4	4
20	$1.2 \cdot 10^{-3}$	18.9	32	$1.2 \cdot 10^{-3}$	2.5	5

Table 4: Newton potential of a Gaussian (B). Ranks and accuracy of the input vectors

convolution times and the same accuracies in the data provided by the authors of [9], and use them for comparison). Note that the accuracies of $\boldsymbol{x}$ and $\boldsymbol{y}$ we obtained initially are enough to find the proper values of the truncation parameter $\varepsilon^{\boldsymbol{x}} = \varepsilon^{\boldsymbol{y}}$ for the series (A) and are far sufficient for the series (B).

Figure 2(a), Figure 3(a) and Figure 4(a) present the accuracy δ of the result $\boldsymbol{z}$, effective rank of $\boldsymbol{z}$ and the convolution time vs. d, respectively, for both the series. The same data along with the accuracy estimate δ_{est} and the accuracy parameter ε of inexact convolution are given in numbers in Table 5 (A) and Table 6 (B).

As we can see from the results, the accuracy estimate of Remark 6.1 is achieved by Algorithm 5.2 (for the series (A) the slope is -2.00 in Figure 2(b)). This is not the case for the QTT-FFT convolution algorithm, the accuracy given for which is also the best possible for the method and is $\mathcal{O}\left(\frac{1}{n}\right)$. In the series (A), the DMRG version with on-the-fly truncation proves to be faster than exact

d	ε	$r^{\boldsymbol{z}}_{\text{eff}}$	$r^{\boldsymbol{z}}_{\max}$	δ	δ_{est}	time
	exact					
10		75.0	124	$1.3 \cdot 10^{-3}$	$4.0 \cdot 10^{-2}$	**0.02**
12		175.0	528	$8.2 \cdot 10^{-5}$	$2.4 \cdot 10^{-3}$	**0.07**
14		311.2	882	$5.1 \cdot 10^{-6}$	$1.5 \cdot 10^{-4}$	**0.22**
16		473.4	1408	$3.2 \cdot 10^{-7}$	$9.4 \cdot 10^{-6}$	**0.61**
18		614.1	1856	$2.1 \cdot 10^{-8}$	$5.9 \cdot 10^{-7}$	**1.31**
20		854.9	2682	$1.3 \cdot 10^{-9}$	$3.7 \cdot 10^{-8}$	**2.78**
	exact + truncation					
10	$1.0 \cdot 10^{-4}$	18.3	29	$1.3 \cdot 10^{-3}$	$4.0 \cdot 10^{-2}$	**0.11**
12	$5.0 \cdot 10^{-6}$	25.7	41	$8.2 \cdot 10^{-5}$	$2.4 \cdot 10^{-3}$	**1.26**
14	$5.0 \cdot 10^{-7}$	31.7	54	$5.2 \cdot 10^{-6}$	$1.5 \cdot 10^{-4}$	**7.66**
16	$5.0 \cdot 10^{-8}$	38.2	69	$3.3 \cdot 10^{-7}$	$9.5 \cdot 10^{-6}$	**29.43**
18	$5.0 \cdot 10^{-9}$	44.2	84	$2.2 \cdot 10^{-8}$	$5.9 \cdot 10^{-7}$	**69.71**
20	$5.0 \cdot 10^{-10}$	50.5	103	$1.4 \cdot 10^{-9}$	$3.7 \cdot 10^{-8}$	**213.59**
	DMRG					
10	$1.0 \cdot 10^{-4}$	19.6	28	$1.3 \cdot 10^{-3}$	$4.0 \cdot 10^{-2}$	**0.29**
12	$5.0 \cdot 10^{-6}$	28.2	43	$8.2 \cdot 10^{-5}$	$2.4 \cdot 10^{-3}$	**1.02**
14	$5.0 \cdot 10^{-7}$	37.0	58	$5.2 \cdot 10^{-6}$	$1.5 \cdot 10^{-4}$	**3.42**
16	$5.0 \cdot 10^{-8}$	43.9	72	$3.3 \cdot 10^{-7}$	$9.4 \cdot 10^{-6}$	**7.87**
18	$5.0 \cdot 10^{-9}$	51.7	88	$2.1 \cdot 10^{-8}$	$6.0 \cdot 10^{-7}$	**18.05**
20	$5.0 \cdot 10^{-10}$	61.6	108	$1.4 \cdot 10^{-9}$	$3.7 \cdot 10^{-8}$	**45.11**

Table 5: Newton potential of a Gaussian (A). Convolution time is given in seconds

convolution with subsequent truncation of the result, as long as QTT ranks are high enough in the case of the best accuracies, mostly due to the not so good QTT structure of f. This changes when we switch to the series (B): in this case the DMRG version is slower than exact one with truncation. However, the difference of the two is not as remarkable of them and the QTT-FFT convolution algorithm, the latter being from 30 to 150 times slower.

In the series (A) we observe roughly the dependencies $r^{\boldsymbol{x}}_{\text{eff}} \sim d^{2.2}$ and $r^{\boldsymbol{y}}_{\text{eff}} \sim d^{1.8}$ for the input ($\varepsilon \sim 10^{-d}$), while for the output ($\delta \sim 10^{-2d}$) effective rank changes as $r^{\boldsymbol{z}}_{\text{eff}} \sim d^{3.2}$ with no rank truncation and $r^{\boldsymbol{z}}_{\text{eff}} \sim d^{1.5}$ with rank truncation. The latter reflects that the explicit representation of convolution in terms of $\boldsymbol{x}$ and $\boldsymbol{y}$ instead of $\boldsymbol{x}\boldsymbol{y}'$ may really have excessive ranks, as we mentioned in Remark 4.8, and the convolution result is to be compressed even for the best accuracies and in spite of that the input vectors cannot be truncated separately without introducing an extra error in the output. For the exact and DMRG versions we see that the convolution time is $\sim d^{7.3}$ (with different constant factors), but truncation of already computed $\boldsymbol{z}$ with $r^{\boldsymbol{z}}_{\text{eff}} \sim d^{3.2}$ is really expensive and requires as much as $\sim d^{11.0}$ operations. However, for all the three versions we may remark the complexity $\mathcal{O}\left(\log^{\alpha} N\right)$ logarithmic w. r. t. the problem size $N = n^3 = 2^{3d}$ in the series (A), and the corresponding constants allow our methods to outperform the QTT-FFT-based convolution in the series (B).

7 Conclusion

We have presented explicitly QTT structure of the multilevel Toeplitz matrix generated by a QTT-structured vector. This relation is established in the form of matrix-vector multiplication of the generator and a proper “structuring” tensor in the QTT format (Theorem 4.6). Similar result has been shown for a matrix-vector product of such a matrix and another QTT-structured vector (Theorem 4.7). Bounds for QTT ranks of the output follow immediately and numerically prove to be sharp.

d	ε	$r^{\boldsymbol{z}}_{\text{eff}}$	$r^{\boldsymbol{z}}_{\max}$	δ	δ_{est}	time	time *	δ_{est} *
	exact						QTT-FFT conv.	
10		22.7	36	$3.8 \cdot 10^{-2}$	$2.8 \cdot 10^{-1}$	**0.009**	**2.9**	$2.9 \cdot 10^{-1}$
12		27.6	56	$2.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	**0.011**	**4.7**	$1.2 \cdot 10^{-1}$
14		32.2	64	$1.9 \cdot 10^{-2}$	$3.7 \cdot 10^{-2}$	**0.013**	**6.6**	$3.7 \cdot 10^{-2}$
16		45.8	96	$6.9 \cdot 10^{-3}$	$8.8 \cdot 10^{-3}$	**0.018**	**8.8**	$9.9 \cdot 10^{-3}$
18		66.5	144	$1.9 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$	**0.027**	**11.2**	$2.5 \cdot 10^{-3}$
20		93.5	250	$6.3 \cdot 10^{-4}$	$5.7 \cdot 10^{-4}$	**0.042**	**13.8**	$5.9 \cdot 10^{-4}$
	exact + truncation						QTT-FFT conv.	
10	$3.0 \cdot 10^{-2}$	6.0	9	$4.0 \cdot 10^{-2}$	$2.8 \cdot 10^{-1}$	**0.018**	**2.9**	$2.9 \cdot 10^{-1}$
12	$1.0 \cdot 10^{-2}$	6.8	10	$2.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	**0.026**	**4.7**	$1.2 \cdot 10^{-1}$
14	$3.0 \cdot 10^{-3}$	8.9	14	$1.9 \cdot 10^{-2}$	$3.7 \cdot 10^{-2}$	**0.040**	**6.6**	$3.7 \cdot 10^{-2}$
16	$6.5 \cdot 10^{-4}$	12.0	20	$6.9 \cdot 10^{-3}$	$9.8 \cdot 10^{-3}$	**0.076**	**8.8**	$9.9 \cdot 10^{-3}$
18	$1.5 \cdot 10^{-4}$	15.2	27	$1.9 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$	**0.176**	**11.2**	$2.5 \cdot 10^{-3}$
20	$3.0 \cdot 10^{-5}$	19.1	38	$6.3 \cdot 10^{-4}$	$5.8 \cdot 10^{-4}$	**0.429**	**13.8**	$5.9 \cdot 10^{-4}$
	DMRG						QTT-FFT conv.	
10	$5.0 \cdot 10^{-2}$	7.6	9	$4.0 \cdot 10^{-2}$	$2.9 \cdot 10^{-1}$	**0.07**	**2.9**	$2.9 \cdot 10^{-1}$
12	$2.5 \cdot 10^{-2}$	9.7	12	$2.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	**0.11**	**4.7**	$1.2 \cdot 10^{-1}$
14	$5.0 \cdot 10^{-3}$	12.1	14	$1.9 \cdot 10^{-2}$	$3.6 \cdot 10^{-2}$	**0.18**	**6.6**	$3.7 \cdot 10^{-2}$
16	$4.7 \cdot 10^{-4}$	15.4	22	$6.9 \cdot 10^{-3}$	$9.5 \cdot 10^{-3}$	**0.29**	**8.8**	$9.9 \cdot 10^{-3}$
18	$2.5 \cdot 10^{-4}$	19.4	26	$1.9 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$	**0.56**	**11.2**	$2.5 \cdot 10^{-3}$
20	$3.0 \cdot 10^{-5}$	23.3	37	$6.3 \cdot 10^{-4}$	$5.6 \cdot 10^{-4}$	**0.88**	**13.8**	$5.9 \cdot 10^{-4}$

Table 6: Newton potential of a Gaussian (B). Convolution time is given in seconds. The two columns marked with * present convolution times and accuracy estimates of the QTT-FFT convolution algorithm [9]

A method (Algorithm 5.2) of multidimensional convolution (Toeplitz matrix-vector multiplication) in the QTT format, exploiting the explicit QTT structure presented, has been proposed. Several versions were considered: exact convolution, exact convolution with subsequent QTT truncation and inexact convolution with on-the-fly QTT truncation, the latter being based on the DMRG approach to matrix-vector multiplication in the QTT format. Numerical experiments in 1D and 3D show that the convolution method proposed is efficient in handling large-scale data and outperforms the FFT-based convolution (in time) and the QTT-FFT-based convolution (in both time and accuracy).

Computation of the Newton potential of a narrow Gaussian with the use of the method proposed shows that very fast and accurate convolution with the complexity scaling logarithmically ($\log^\alpha$) w. r. t. the problem size is achievable by employing the QTT format for nonlinear approximation, and highlights that this format may be beneficial for computations with functions with singularities on simple uniform tensor grids, while sophisticated adaptive methods are usually brought into play to deal with such functions.

Apart from numerous applications in scientific computing and signal and image processing, the results of this paper may be useful in view of handling the algebra of Toeplitz matrices and multiplication of polynomials in the QTT format. It is worth noting that the theoretical results of the paper are independent of the field under consideration ($\mathbb{R}$ or $\mathbb{C}$, for other fields a proper TT arithmetics might be needed) and the convolution algorithm proposed can be applied straightforwardly to the complex-valued as well as real-valued data.

References

[1] *T. G. Kolda, B. W. Bader*. Tensor Decompositions and Applications // *SIAM Review*. 2009, September. V. 51, No. 3. P. 455–500. DOI: 10.1.1.153.2059. http://citeseerx.ist.psu.edu/

viewdoc/download?doi=10.1.1.153.2059&rep=rep1&type=pdf. 1, 3

[2] *L. De Lathauwer*. A survey of tensor methods // *Proceedings of the 2009 IEEE International Symposium on Circuits and Systems*. 2009, May. — P. 2773 -2776. ftp://ftp.esat.kuleuven.ac.be/pub/pub/SISTA/delathauwer/reports/ldl-09-34.pdf. 1, 3

[3] *B. N. Khoromskij*. Tensors-structured Numerical Methods in Scientific Computing: Survey on Recent Advances: Preprint 21: Max-Planck-Institut für Mathematik in den Naturwissenschaften, 2010. http://www.mis.mpg.de/publications/preprints/2010/prepr2010-21.html. 1, 3

[4] *B. N. Khoromskij*. Fast and accurate tensor approximation of a multivariate convolution with linear scaling in dimension // *J. Comput. Appl. Math.* 2010, October. V. 234. P. 3122–3139. DOI: 10.1016/j.cam.2010.02.004. http://dx.doi.org/10.1016/j.cam.2010.02.004. 1

[5] *J.-P. Calliess, M. Mai, S. Pfeiffer*. On the computational benefit of tensor separation for high-dimensional discrete convolutions // *Multidimensional Systems and Signal Processing*. 2010. P. 1-25. DOI: 10.1007/s11045-010-0131-2. http://www.springerlink.com/content/m51lh4121313214r. 1, 4

[6] *I. Oseledets*. Approximation of matrices with logarithmic number of parameters // *Doklady Mathematics*. 2009. V. 80. P. 653–654. DOI: 10.1134/S1064562409050056. http://dx.doi.org/10.1134/S1064562409050056. 1, 2, 4

[7] *B. N. Khoromskij*. $\mathcal{O}(d \log N)$-Quantics Approximation of N-d Tensors in High-Dimensional Numerical Modeling: Preprint 55: Max-Planck-Institut für Mathematik in den Naturwissenschaften, 2009. http://www.mis.mpg.de/publications/preprints/2009/prepr2009-55.html. 1, 2, 4

[8] *I. V. Oseledets*. Approximation of $2^d \times 2^d$ matrices using tensor decomposition // *SIAM Journal on Matrix Analysis and Applications*. 2010. V. 31, No. 4. P. 2130–2145. DOI: 10.1137/090757861. http://link.aip.org/link/?SML/31/2130/1. 1, 2, 4

[9] *S. Dolgov, B. N. Khoromskij, D. Savostyanov*. Multidimensional Fourier transform in logarithmic complexity using QTT approximation: Preprint 18: Max-Planck-Institut für Mathematik in den Naturwissenschaften, 2011. http://www.mis.mpg.de/publications/preprints/2011/prepr2011-18.html. 1, 4, 18, 19, 22, 24, 25, 27

[10] *B. Khoromskij*. $\mathcal{O}(d \log N)$-Quantics Approximation of N-d Tensors in High-Dimensional Numerical Modeling // *Constructive Approximation*. 2011. P. 1-24. DOI: 10.1007/s00365-011-9131-1, 10.1007/s00365-011-9131-1. http://dx.doi.org/10.1007/s00365-011-9131-1. 3

[11] *I. V. Oseledets*. Constructive representation of functions in tensor formats: Preprint 4: Institute of Numerical Mathematics of RAS, 2010, August. http://pub.inm.ras.ru/pub/inmras2010-04.pdf. 3

[12] *V. A. Kazeev, B. N. Khoromskij*. On explicit QTT representation of Laplace operator and its inverse: Preprint 75: Max-Planck-Institut für Mathematik in den Naturwissenschaften, 2010. http://www.mis.mpg.de/publications/preprints/2010/prepr2010-75.html. 3, 5

[13] *G. Heinig, K. Rost*. Algebraic methods for Toeplitz-like matrices and operators. — Mathematical Research, Vol. 19. Berlin: Akademie-Verlag. 212 p. M 28.00; licenced edition: Operator Theory: Advances and Applications, Vol. 13. Basel - Boston - Stuttgart: Birkhäuser Verlag. 212 p. DM 64.00 , 1984. http://www.getcited.org/pub/102915186. 3

[14] *V. V. Voevodin, E. E. Tyrtyshnikov*. Computational processes with Toeplitz matrices (in Russian). — Nauka, 1987. http://books.google.com/books?id=pf3uAAAAMAAJ. 3

[15] *A. Böttcher, B. Silberman*. Introduction to large truncated Toeplitz Matrices. — Berlin-Heidelberg-New York: Springer, 1999. http://books.google.com/books?id=3Dd0KnravR8C. 3

[16] *R. H. Chan, M. K. Ng*. Conjugate Gradient Methods for Toeplitz Systems // *SIAM Review*. 1996. V. 38, no. 3. P. 427–482. http://www.jstor.org/stable/2132496. 3

[17] *T. Kailath, S.-Y. Kung, M. Morf*. Displacement ranks of matrices and linear equations // *Journal of Mathematical Analysis and Applications*. 1979. V. 68, No. 2. P. 395–407. DOI: 10.1016/0022-247X(79)90124-0. http://www.sciencedirect.com/science/article/B6WK2-4CRM702-17V/2/8b466243f75860950d3cf564c5fdde8f. 3

[18] *V. Olshevsky, I. Oseledets, E. Tyrtyshnikov*. Tensor properties of multilevel Toeplitz and related matrices // *Linear Algebra and its Applications*. 2006. V. 412, No. 1. P. 1 - 21. DOI: 10.1016/j.laa.2005.03.040. http://www.sciencedirect.com/science/article/B6V0R-4H10BT5-1/2/30ecb163a954e5ee5357b7de770b49b5. 3

[19] *V. Olshevsky, I. Oseledets, E. Tyrtyshnikov*. Superfast Inversion of Two-Level Toeplitz Matrices Using Newton Iteration and Tensor-Displacement Structure // Recent Advances in Matrix and Operator Theory / Ed. by J. A. Ball, Y. Eidelman, J. W. Helton et al. — Birkhäuser Basel, 2008. — V. 179 of *Operator Theory: Advances and Applications*. — P. 229-240. http://dx.doi.org/10.1007/978-3-7643-8539-2_14. 3

[20] *B. N. Khoromskij*. Fast and accurate tensor approximation of a multivariate convolution with linear scaling in dimension // *Journal of Computational and Applied Mathematics*. 2010. V. 234, No. 11. P. 3122 - 3139. DOI: 10.1016/j.cam.2010.02.004, Numerical Linear Algebra, Internet and Large Scale Applications. http://www.sciencedirect.com/science/article/pii/S0377042710000750. 3, 4, 19, 22

[21] *B. Khoromskij*. On tensor approximation of Green iterations for Kohn-Sham equations // *Computing and Visualization in Science*. 2008. V. 11. P. 259-271. DOI: 10.1007/s00791-008-0097-x. http://dx.doi.org/10.1007/s00791-008-0097-x. 4

[22] *H.-J. Flad, B. Khoromskij, D. Savostyanov, E. Tyrtyshnikov*. Verification of the cross 3D algorithm on quantum chemistry data // *Russian Journal of Numerical Analysis and Mathematical Modelling*. 2008, August. V. 23, No. 4. P. 210–220. DOI: 10.1515/RJNAMM.2008.020. http://www.reference-global.com/doi/abs/10.1515/RJNAMM.2008.020. 4, 20

[23] *B. Khoromskij, V. Khoromskaia, S. Chinnamsetty, H.-J. Flad*. Tensor decomposition in electronic structure calculations on 3D Cartesian grids // *Journal of Computational Physics*. 2009. V. 228, No. 16. P. 5749 - 5762. DOI: 10.1016/j.jcp.2009.04.043. http://www.sciencedirect.com/science/article/pii/S0021999109002356. 4, 19, 20

[24] *V. Khoromskaia*. Numerical solution of the Hartree-Fock equation by multilevel tensor-structured methods: Ph.D. thesis / Technische Universität Berlin. — 2010. http://opus.kobv.de/tuberlin/volltexte/2011/2948/. 4, 20

[25] *B. N. Khoromskij, V. Khoromskaia, H.-J. Flad*. Numerical Solution of the Hartree–Fock Equation in Multilevel Tensor-Structured Format // *SIAM Journal on Scientific Computing*. 2011. V. 33, No. 1. P. 45–65. DOI: 10.1137/090777372. http://link.aip.org/link/?SCE/33/45/1. 4, 20

[26] *R. Bellman*. Adaptive Control Processes: A Guided Tour. — Princeton, NJ: Princeton University Press, 1961. 4

[27] *W. Hackbusch*. Efficient convolution with the Newton potential in d dimensions // *Numerische Mathematik*. 2008. V. 110. P. 449–489. DOI: 10.1007/s00211-008-0171-9. http://dx.doi.org/10.1007/s00211-008-0171-9. 4

[28] *W. Hackbusch, K. K. Naraparaju, J. Schneider*. On the efficient convolution with the Newton potential // *Journal of Numerical Mathematics*. 2010. V. 17, No. 4. P. 257–280. DOI: 10.1515/JNUM.2010.013. http://www.reference-global.com/doi/abs/10.1515/JNUM.2010.013. 4

[29] *V. de Silva, L.-H. Lim*. Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem // *SIAM Journal on Matrix Analysis and Applications*. 2008. V. 30, No. 3. P. 1084–1127. 4

[30] *I. Oseledets, E. Tyrtyshnikov*. Recursive decomposition of multidimensional tensors // *Doklady Mathematics*. 2009. V. 80. P. 460–462. 10.1134/S1064562409040036. http://dx.doi.org/10.1134/S1064562409040036. 4

[31] *I. Oseledets*. A new tensor decomposition // *Doklady Mathematics*. 2009. V. 80. P. 495–496. DOI: 10.1134/S1064562409040115. http://dx.doi.org/10.1134/S1064562409040115. 4

[32] *I. V. Oseledets, E. E. Tyrtyshnikov*. Breaking the curse of dimensionality, or how to use SVD in many dimensions // *SIAM Journal on Scientific Computing*. 2009, October. V. 31, No. 5. P. 3744–3759. DOI: 10.1137/090748330. http://epubs.siam.org/sisc/resource/1/sjoce3/v31/i5/p3744_s1. 4

[33] *I. V. Oseledets*. Tensor Train decomposition // *To appear in SIAM Journal on Scientific Computing*. 4

[34] *S. R. White*. Density matrix formulation for quantum renormalization groups // *Phys. Rev. Lett.* 1992, November. V. 69, No. 19. P. 2863–2866. DOI: 10.1103/PhysRevLett.69.2863. http://link.aps.org/doi/10.1103/PhysRevLett.69.2863. 4

[35] *S. R. White*. Density-matrix algorithms for quantum renormalization groups // *Phys. Rev. B*. 1993, October. V. 48, No. 14. P. 10345–10356. DOI: 10.1103/PhysRevB.48.10345. http://link.aps.org/doi/10.1103/PhysRevB.48.10345. 4

[36] *F. Verstraete, D. Porras, J. I. Cirac*. Density Matrix Renormalization Group and Periodic Boundary Conditions: A Quantum Information Perspective // *Phys. Rev. Lett.* 2004, November. V. 93, No. 22. P. 227205. DOI: 10.1103/PhysRevLett.93.227205. http://link.aps.org/doi/10.1103/PhysRevLett.93.227205. 4

[37] *G. Vidal*. Efficient Classical Simulation of Slightly Entangled Quantum Computations // *Phys. Rev. Lett.* 2003, October. V. 91, No. 14. P. 147902. DOI: 10.1103/PhysRevLett.91.147902. http://link.aps.org/doi/10.1103/PhysRevLett.91.147902. 4

[38] *E. E. Tyrtyshnikov*. Tensor approximations of matrices generated by asymptotically smooth functions // *Sbornik: Mathematics*. 2003. V. 194, No. 5. P. 941–954. DOI: 10.1070/SM2003v194n06ABEH000747. http://iopscience.iop.org/1064-5616/194/6/A09. 4

[39] *W. Hackbusch, S. Kühn*. A New Scheme for the Tensor Representation // *Journal of Fourier Analysis and Applications*. 2009. V. 15. P. 706–722. 10.1007/s00041-009-9094-9. http://dx.doi.org/10.1007/s00041-009-9094-9. 4

[40] *L. Grasedyck*. Polynomial Approximation in Hierarchical Tucker Format by Vector-Tensorization: Preprint 308: Institut für Geometrie und Praktische Mathematik, RWTH Aachen, 2010, April. http://www.igpm.rwth-aachen.de/Download/reports/pdf/IGPM308_k.pdf. 4

[41] *W. Hackbusch*. Tensorisation of vectors and their efficient convolution // *Numerische Mathematik*. 2011. P. 1–24. DOI: 10.1007/s00211-011-0393-0, 10.1007/s00211-011-0393-0. http://www.springerlink.com/content/64846p36566487p3. 4

[42] *I. V. Oseledets*. DMRG approach to fast linear algebra in the TT-format: Preprint: Hausdorff Research Institute for Mathematics, 2011, July. http://www.hausdorff-research-institute.uni-bonn.de/files/preprints/AnalysisandNumerics/mvk.pdf. 17, 18

[43] *T. Helgaker, P. R. Taylor*. Gaussian basis sets and molecular integrals. In: Modern Electronic Structure Theory. Part II // Ed. by D. R. Yarkony. — Advanced Series in Physical Chemistry. World Scientific, 1995, June. — P. 725–856. http://books.google.com/books?id=Gt4pnp-UFhUC. 20, 23

[44] *L. Tucker*. Some mathematical notes on three-mode factor analysis // *Psychometrika*. 1966. V. 31. P. 279–311. DOI: 10.1007/BF02289464. http://dx.doi.org/10.1007/BF02289464. 20

[45] *L. D. Lathauwer, B. D. Moor, J. Vandewalle*. A Multilinear Singular Value Decomposition // *SIAM Journal on Matrix Analysis and Applications*. 2000. V. 21, No. 4. P. 1253-1278. DOI: 10.1137/S0895479896305696. http://link.aip.org/link/?SML/21/1253/1. 20

[46] *W. Hackbusch, B. Khoromskij*. Low-rank Kronecker-product Approximation to Multi-dimensional Nonlocal Operators. Part I. Separable Approximation of Multi-variate Functions // *Computing*. 2006. V. 76. P. 177-202. 10.1007/s00607-005-0144-0. http://dx.doi.org/10.1007/s00607-005-0144-0. 23