Superfast inversion of two-level Toeplitz matrices using Newton iteration and tensor-displacement structure

To blessed memory of Georg Heinig

Vadim Olshevsky,

University of Connecticut

Ivan Oseledets, Eugene Tyrtyshnikov [1]

Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street, 8, Moscow 119991

Abstract

A fast approximate inversion algorithm is proposed for two-level Toeplitz matrices (block Toeplitz matrices with Toeplitz blocks). It applies to matrices that can be sufficiently accurately approximated by matrices of low Kronecker rank and involves a new class of tensor-displacement-rank structured (TDS) matrices. The complexity depends on the prescribed accuracy and typically is $o(n)$ for matrices of order n.

1 Introduction

Dense matrices arise, for example, in numerical solution of multidimensional integral equations, their approximate inverses are often of interest either themselves or as preconditioners in iterative methods, and the size of matrices occurs to be about a few hundred of thousands or even millions. These cases are

Email addresses: olshevsky@math.uconn.edu (Vadim Olshevsky), ivan@bach.inm.ras.ru (Ivan Oseledets), tee@inm.ras.ru (Eugene Tyrtyshnikov).

[1] Supported by the Russian Fund of Basic Research (grant 05-01-00721) and a Priority Research Grant of the Department of Mathematical Sciences of the Russian Academy of Sciences.

not very easy to handle. The standard Gaussian elimination has the $O(n^3)$ complexity and is unacceptable. Even a method with $O(n^2)$ complexity (an obvious lower bound) is still too slow for matrices on this scale. Luckily, in many cases the matrices possess some structure suggesting a way to make them tractable.

If A is a nonsingular Toeplitz matrix ($a_{ij} = a_{i-j}$), then all the entries of A^{-1} can be computed in $O(n^2)$ operations [13]. It is ever more important that A^{-1} can be expressed by the Gohberg–Semencul formula [3] through some $O(n)$ parameters so that it can be multiplied by a vector in $O(n \log n)$ operations. A tremendous impact of this formula on the field of structured matrices and numerical algorithms was systematically presented in the remarkable book by G. Heinig and K. Rost [5]. A direct but nontrivial generalization to block Toeplitz matrices is the Gohberg–Heinig formula [2].

In this paper we consider two-level Toeplitz matrices, which are block Toeplitz matrices with Toeplitz blocks. If p is simultaneously the block size and the size of blocks, then $n = p^2$ and such a matrix is defined by $O(n)$ parameters. In this case the Gohberg–Heinig formula contains as many as $O(p^3) = O(n^{3/2})$ parameters, which is viewed as too many, when compared with $O(n)$. A better approach can be one that we outlined and started to develop in [9]. However, it applies only to those two-level Toeplitz matrices that are of low tensor (Kronecker) rank. As a nice consequence of this combination of Toeplitz and tensor structure, such matrices are determined by $O(\sqrt{n})$ parameters, the same is expected from their approximate inverse matrices and may (and does, as we show) result in the $o(n)$ complexity. Luckily again, this special subclass of two-level Toeplitz matrices seems to cover all practically interesting matrices.

We will make use of the following iterative method attributed to Hotelling [6] and Schulz [12]:

$$X_i = 2X_{i-1} - X_{i-1}AX_{i-1}, \quad i = 0, 1, \ldots, \tag{1}$$

where X_0 is some initial approximation to A^{-1}. Since $I - AX_i = (I - AX_{i-1})^2$, the iterations (1) converge quadratically, provided that $||I - AX_0|| < 1$. This method is a special form of the Newton method for nonlinear equations and referred to as *Newton iteration*. It has some nice properties such as numerical stability and ease for parallel computations. All the same, each iteration requires two matrix multiplications, which is expensive for general matrices.

In order to perform the Newton iteration in a fast way, we need the following two ingredients:

- A fast matrix-by-matrix procedure;
- A method to preserve structure.

The first means that X_k and A must hold on some structure to facilitate the computation of matrix products. However, if the X_k do not belong to a commutative algebra (circulants, diagonal matrices etc), every next iterate X_{k+1} might be "less structured". As a consequence, the matrix-by-matrix complexity grows with every iteration. In order to slow down this growth, we should preserve the structure by "brute force" — using a method to substitute computed iterates with some approximations by "better structured matrices". We introduce a truncation operator $R(X)$ acting on $n \times n$ matrices as kind of a nonlinear projector. Then, the Newton iteration with approximations (truncations) reads

$$X_i = R(2X_{i-1} - X_{i-1}AX_{i-1}). \quad i = 0, 1, \ldots. \tag{2}$$

The Newton iteration was successfully applied to matrices with the displacement structure [1,11] and matrices represented as a sum of tensor (Kronecker) products [10]. In the case of low-displacement-rank matrices, V. Pan [11] proved that the quadratic convergence is maintained even after truncations. Then, it was discovered in [10] that the latter property holds true for many useful structures rather than one considered in [11]. A pretty general formulation stemming from [10] is given in [4].

Theorem 1.1 *Suppose that* $||(R(X) - A^{-1}|| \leq M||X - A^{-1}||$ *for all* X*. Then for any initial guess* X_0 *sufficiently close to* A^{-1}*, the truncated Newton iterates (2) converge quadratically:*

$$||A^{-1} - X_k|| \leq (1 + M)\,||A||||A^{-1} - X_{k-1}||^2, \qquad k = 1, 2, \ldots\ .$$

Now, with this encouraging result, we are going to propose an algorithm for computing an approximate inverse to a given two-level Toeplitz matrix. Our main idea is to combine two efficient matrix representations using the low-Kronecker-rank and low-displacement-rank properties. Thus, we introduce a new matrix format — the TDS format (tensor displacement structure), and therefore assume that A and A^{-1} should be in the TDS format, at least approximately. A rigorous theory behind this assumption is still lacking; however, all of our numerical experiments on various matrices show that the complexity of the proposed algorithm is $O(\sqrt{n}\log n)$.

The paper is organized as follows.

In section 1 we define the TDS format and the transformation of a two-level Toeplitz matrix into this format. In Section 2 we describe all the basic matrix operations in the TDS format and propose a *fast recompression procedure* (in other words, define the operator R).

In Section 3 we discuss the Newton iteration with approximations and its modification which speeds up the computations dramatically. Also, we suggest

a method for efficient selection of the initial guess X_0. In Section 4 we present some numerical experiments.

2 The TDS format

Below we recall a general notation of multilevel matrices introduced in [14] and the displacement rank constructions presented in [5] as a far-reaching development of the definition introduced first in [7].

Definition 2.1 *A matrix T is considered as two-level with the size-vector (n_1, n_2) if it contains $n_1 \times n_1$ blocks and each block is of size $n_2 \times n_2$. Such a matrix is called* two-level Toeplitz matrix *if*

$$T = [a(\mathbf{i} - \mathbf{j})], \tag{3}$$

where $\mathbf{i} = (i_1, i_2)$ and $\mathbf{j} = (j_1, j_2)$ define the place of the element in the two-level matrix: (i_1, j_1) specifies the block position and (i_2, j_2) does the element location inside the block.

Definition 2.2 *The operator L is said to be of Sylvester type if*

$$L(M) = \nabla_{A,B}(M) = AM - MB \tag{4}$$

and of Stein type if

$$L(M) = \triangle_{A,B}(M) = M - AMB. \tag{5}$$

The value $\alpha \equiv \mathrm{rank}(L(M))$ is called the displacement rank of M. Any $n \times \alpha$ matrices G and H from the skeleton decomposition

$$L(M) = GH^\top$$

are called the generators of M. A matrix defined by its generators is referred to as a displacement-structured matrix. By the very definition, displacement ranks and generators of a matrix depend on the choice of the displacement operator L.

We will use the Stein type operators. The Toeplitz matrices can be associated with the displacement operators $Z_a, Z_b^\top$, where

$$Z_a = Z + ae_0e_{n-1}^\top, \qquad Z_b = Z + be_0e_{n-1}^\top,$$

Z is a unit lower shift matrix and a, b are some scalars. Let $\triangle_{Z_a, Z_b^\top}(M) = GH^\top$ and $G = [g_1, ..., g_\alpha]$, $H = [h_1, ..., h_\alpha]$. Then

$$(1 - ab)M = \sum_{j=1}^{\alpha} Z_a(g_j) Z_b^\top(h_j). \tag{6}$$

Here, $Z_a(g)$ and $Z_b(h)$ are defined as follows. Let c be a scalar and v a vector; then $Z_c(v)$ is a Toeplitz matrix with the entries

$$(Z_c(v))_{ij} = \begin{cases} v_{i-j}, & i - j \geq 0, \\ c\, v_{n+i-j}, & i - j < 0. \end{cases}$$

If M is nonsingular, then M^{-1} can be expressed by a formula of the same type as (6), considered in this case as one of possible generalizations of the Gohberg–Semencul formula to Toeplitz-like matrices. Both in the latter formula and in (6), a matrix is the sum of special Toeplitz matrices belonging to some algebras; however, the Gohberg–Semencul formula and (6) use different algebras. If M is a Toeplitz matrix then $\alpha \leq 2$.

Definition 2.3 *A matrix A is said to be in the tensor format of the tensor rank r, if*

$$A = \sum_{k=1}^{r} A_k^1 \otimes A_k^2. \tag{7}$$

Given a two-level matrix A, we can try to approximate it by a low-tensor-rank matrix. Let

$$\mathcal{V}_{\mathbf{n}}(A) = [b_{(i_1, j_1)(i_2, j_2)}]$$

be a two-level matrix with the size-vectors (n_1, n_1) and (n_2, n_2), and define it by the rule

$$b_{(i_1, j_1)(i_2, j_2)} = a_{(i_1, i_2)(j_1, j_2)}.$$

Then, as is readily seen, the tensor rank of A is equal to the rank of $\mathcal{V}_{\mathbf{n}}(A)$. Moreover,

$$||A - A_r||_F = ||\mathcal{V}_{\mathbf{n}}(A) - \mathcal{V}_{\mathbf{n}}(A_r)||_F,$$

which reduces the problem of optimal tensor approximation to the problem of optimal lower-rank approximation. The latter can be solved using the SVD or the Lanzos bidiagonalization algorithm. However, in the case of two-level Toeplitz matrices we can solve this problem much easier [8] (for more general constructions see [9]).

Given $T = [a(\mathbf{i} - \mathbf{j})]$, we compose a smaller matrix

$$W(A) = [a_{\mu\nu}], \quad 1 - n_1 \leq \mu \leq n_1 - 1,\ 1 - n_2 \leq \nu \leq n_2 - 1, \tag{8}$$

construct an optimal rank-r approximation

$$W(A) \approx \sum_{k=1}^{r} u_k v_k^\top,$$

$$U^k = [u^k_{i_1 - j_1}], \quad 0 \le i_1, j_1 \le n_1 - 1,$$
$$V^k = [v^k_{i_2 - j_2}], \quad 0 \le i_2, j_2 \le n_2 - 1,$$

and finish with the tensor approximation of the form

$$T \;\approx\; T_r = \sum_{k=1}^{r} U^k \otimes V^k. \tag{9}$$

It is proved that this is an optimal tensor-rank-r approximation to T in the Frobenius norm. The computational cost is that of finding a low-rank approximation to the matrix of size $(2n_1-1)\times(2n_2-1)$. Remarkably, the tensor factors are themselves Toeplitz matrices. A crucial parameter defining the complexity is the tensor rank r. It depends on the prescribed approximation accuracy and is directly related to the properties of the symbol (generating function) of T. Some upper estimates on r were proposed in [9] for asymptotically smooth symbols.

It is proved in [9] that a two-level Toeplitz matrix with an approximately separable symbols can be approximated by a sum of tensor products of Toeplitz matrices. Now we embed this format into a more general one which suits better to approximate the corresponding inverse matrices.

Definition 2.4 *A two-level matrix A is said to be in the TDS (tensor-displacement structure) format if it is in the tensor format (7) with each factor being a displacement-structured matrix.*

Let r be the tensor rank and s the maximal displacement rank of the factors. Obviously, the TDS format requires a storage of $O(\sqrt{n}rs)$ cells.

3 Matrix arithmetic in the TDS format

3.1 Basic operations in the displacement format

Consider matrices A and B of Toeplitz displacement rank α and β. Then it is is well known that

- a matrix-by-vector product Ax can be computed in $O(\alpha n \log n)$ operations;
- a matrix-by-matrix product AB can be computed in $O(\alpha\beta n \log n)$ operations, with the displacement rank of AB increasing at most to $\alpha + \beta$.

3.2 Basic operations in the tensor format

If two matrices M_1 and M_2 are in the tensor format

$$M^1 = \sum_{i=1}^{r_1} A_i^1 \otimes B_i^1, \quad M^2 = \sum_{i=1}^{r_2} A_i^2 \otimes B_i^2,$$

then the product

$$M^1 M^2 = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} (A_i^1 A_j^2) \otimes (B_i^1 B_j^2) \tag{10}$$

is already in the tensor format. However, it requires a larger storage as the tensor ranks are to be multiplied. The sum of two matrices in the tensor format is also in some tensor format (we even should not do anything — only merge two arrays). But again, the tensor rank grows. Thus, we should find a way to approximate the results of matrix operations by matrices of lower tensor rank. This task can be accomplished very efficiently through an SVD-based procedure called *recompression*. Since the problem of finding a low-tensor-rank approximation to matrix A is equivalent to the problem of finding a low-rank approximation to $\mathcal{V}_{\mathbf{n}}(A)$, we can expoit the following.

Given a low rank matrix $B = UV^\top$, $U, V \in \mathbb{R}^{n \times r}$, we can find $q \leq r$ and matrices $\widetilde{U}, \widetilde{V}^\top \in \mathbb{R}^{n \times q}$ approximating A with the desired accuracy ε:

$$||B - \widetilde{U}\widetilde{V}^\top||_F \leq \varepsilon ||B||_F. \tag{11}$$

All we need is to find the SVD of B. Since B is already in the low-rank format, we proceed as follows:

(1) Find the QR-decomposion of U and V: $U = Q_u R_u, V = Q_v R_v$;

(2) Find the SVD of a $r \times r$ matrix $R_u R_v^\top$: $R_u R_v^\top = U_1 \Sigma V_1^\top$.

Then, $B = (Q_u U_1)\Sigma(Q_v V_1)$ is the SVD of B. Now, take the smallest possible q so that

$$\sigma_{q+1}^2 + ... + \sigma_r^2 \leq \varepsilon ||B||_F.$$

When r is small, the cost of thid method is dominated by the QR-decomposition complexity, which is $O(nr^2)$, and is linear in matrix size. But, recall that the columns of U and V come from the reshaped tensor factors which are stored in the displacement format (to be extracted from the generators). Does it help to perform the recompression faster? The answer is yes, the algorithm being described in the next subsection.

Let us look more closely at the recompression steps. The QR-decomposition can be implemented through the Gramm–Schmidt orthogonalization algorithm applied to the vectors $u_1, ..., u_r$. The orthogonality is defined by the ordinary scalar product $(x, y) = \sum_{k=1}^{n} x_i \overline{y}_i$. Now, instead of working with vectors, we suggest to work directly with their matrix prototypes. The scalar product for matrices is defined as the *Frobenius scalar product*:

$$(A, B)_F = \mathrm{tr}(\mathrm{AB}^*).$$

Other operations required in the Gramm–Schmidt algorithm, which are multiplication by numbers and addition, can be performed directly with matrices. Moreover, employing the displacement structure in these operations leads to the $O(\sqrt{n}\log n)$ complexity. Thus, we should focus on fast calculation of the Frobenius scalar product of two matrices given in the displacement formats.

Given $p \times p$ matrices A and B with displacement ranks α and β, we need to find $\mathrm{tr}(\mathrm{AB}^*)$. First, we calculate AB^*. As we know, that can be done in $O((\alpha+\beta)p\log p)$ operations and the displacement rank of the product does not exceed $\alpha + \beta$. It remains to calculate the trace of a Toeplitz-like matrix. Fortunately, this can be done by a simple formula involving the generators.

Lemma 3.1 *Let C be a $p \times p$ matrix and $\triangle_{Z_a, Z_b^\top}(C) = GH^T$, $G = [g^1, ..., g^\alpha]$, $H = [h^1, ..., h^\alpha]$, where $h^i, g^i \in \mathbb{R}^p$. Then*

$$\mathrm{tr}(C) = \frac{1}{1-ab}\sum_{r=1}^{\alpha}\sum_{k=0}^{p-1} h_k^r g_k^r (p - k + abk). \tag{12}$$

Proof. According to (6), the matrix C can be represented as

$$C = \frac{1}{1-ab}\sum_{j=1}^{\alpha} Z_a(g_j) Z_b^\top(h_j).$$

Therefore,

$$\mathrm{tr}(C) = \frac{1}{1-ab}\sum_{j=1}^{\alpha} \mathrm{tr}(Z_a(g_j) Z_b^\top(h_j)). \tag{13}$$

Each term in the sum (13) is of the form

$$\mathrm{tr}(Z_a(g)Z_b^\top(h)) = \sum_{i=0}^{p-1}(Z_a(g)Z_b^\top(h))_{ii} = \sum_{i=0}^{p-1}\sum_{k=0}^{p-1} Z_a(g)_{ik} Z_b(h)_{ik} =$$

$$= \sum_{i=0}^{p-1}\sum_{k=0}^{i} g_{i-k}h_{i-k} + ab\sum_{i=0}^{p-1}\sum_{k=i+1}^{p-1} g_{p+i-k}h_{p+i-k}.$$

The first summand is transformed as

$$\sum_{i=0}^{p-1}\sum_{k=0}^{i} g_{i-k}h_{i-k} = \sum_{i=0}^{p-1}\sum_{k=0}^{i} g_k h_k = \sum_{k=0}^{p-1} h_k g_k (p-k),$$

and, similarly, the second one is

$$\sum_{i=0}^{p-1}\sum_{k=i+1}^{p-1} g_{p+i-k}h_{p+i-k} = \sum_{k=0}^{p-1} h_k g_k k.$$

3.4 Truncation operator

The truncation operator $R(X)$ can be defined by setting either some bounds on the ranks or accuracy. Fixing the ranks, we find $R_{\rho,s}(X)$ through the following steps:

(1) Find the best tensor-rank-ρ approximation X_ρ to X using the fast recompression algorithm.

(2) Approximate tensor factors by some displacement-rank-s matrices.

It can be verified that such an operator satsfies the conjectures of Theorem 1.1. It follows that the Newton method with the truncation operator $R_{\rho,s}(X)$ retains an important property of quadratic convergence.

However, in practice it is expedient to prescribe the accuracy and let the rank vary. Denote the corresponding operator by R_ε. Formally the steps are the same, but the ranks are no longer constant. The first step ends with the best low-tensor approximation to X satisfying $||X - X_r|| \le \varepsilon ||X||$, the second step produces an approximation with the preset accuracy and smallest possible displacement rank.

4 Newton iteration for approximate inversion of matrices

Let A be in the TDS format. If an initial approximation X_0 to A^{-1} is in the same format, then it can be fastly improved by the iteration (1). The residuals $R_k = I - AX_k$ satisfy $R_{k+1} = R_k^2$, which proves the quadratic convergence of the process provided that the spectral radius of R_0 is less than 1. The initial approximation can be always selected as

$$X_0 = \alpha A^*$$

with some $\alpha > 0$. In this case the estimated number of the operations to achieve accuracy $||A^{-1} - X_k||_2/||A^{-1}||_2 \leq \varepsilon$ is

$$\log_2 (c^2 + 1) + \log_2 \ln \frac{1}{\varepsilon},$$

where c is the spectral condition number of matrix A. For ill-conditioned matrices, the cost is dominated by $\log_2 (c^2 + 1)$.

4.1 Modified Newton iteration

On each step of the Newton method (2), we replace X_k with $R_\varepsilon(X_k)$, where ε is the accuracy parameter. We can also use a modification [10] that works with approximations much better. Indeed, a typical tensor rank of matrices in our examples is about $10 \div 15$, so each Newton step involves about 200 multiplications of Toeplitz-like matrices with the displacement ranks being typically about 10. Following [10], we consider the following modification of the Newton iteration:

$$X_k \;=\; X_{k-1}(2I - X_{k-1}), \quad Y_k \;=\; Y_{k-1}(2I - X_{k-1}), \qquad k = 1, 2, \ldots , \quad (14)$$

where Y_0 is an initial approximation to A^{-1} and $X_0 = AY_0$ is a nonsingular matrix of which we require that the spectral radius of $I - X_0$ is less than 1. The latter implies

$$\lim_{k\to\infty} X_k = I,$$

and since it is easy to derive from (14) that

$$Y_{k+1}X_{k+1}^{-1} = Y_k X_k^{-1} = \cdots = Y_0 X_0^{-1} = A^{-1},$$

we conclude that

$$\lim_{k\to\infty} Y_k = A^{-1}.$$

In case of general matrices, it is easy to see that (14) is just another way of writing (1). However, in the approximate arithmetic the situation changes dramatically. The modified Newton method with approximations now reads

$$X_k \;=\; R_\varepsilon(X_{k-1}(2I - X_{k-1})), \quad Y_k \;=\; R_\varepsilon(Y_{k-1}(2I - X_{k-1})), \quad k = 1, 2, \ldots . \quad (15)$$

A good argument in favour of this modification is the following. As long as X_k converges to the identity matrix, its tensor rank decreases and, hence, the displacement ranks of the factors become smaller and cause the complexity get down. (This should supposedly hold for any class of structured matrices in which the identity matrix is considered as one with "perfect" structure).

4.2 Selection of the initial approximation

Selection of the initial approximation X_0 to A^{-1} is crucial, in particular for ill-conditioned matrices. The common choice $X_0 = \alpha A^*$ with an appropriate $\alpha > 0$ is ever available, of course, but never good if we want a sufficiently accurate answer. However, we can play with the accuracy parameter ε. In the case of structured matrices it controls both the final accuracy and the truncation accuracy on iterations. Thus, it accounts for the ranks after truncation, and thence the speed of calculations. When the process is "far" from the fast convergence stage, we can carry out the truncation with a much lower accuracy ε. Consequently, the matrix operations become pretty fast in the beginning. On later stages ε must diminish and in the end stay on the level of the desired final accuracy.

This idea was used in [10] for a two-level Toeplitz matrix arising after discretization of a hypersingular integral equation. It can be summarized in the following scheme:

(1) Set $X_0 = \alpha A^*$ and perform the Newton iteration with the truncation accuracy $\delta \gg \varepsilon$. This results in a rough approximation M to the inverse, but the advantage is that the δ-truncated Newton iterations are expected to have a low complexity.

(2) Use the previous approximation M as a new guess to start the Newton iteration with finer accuracy ε.

Of course, this scheme can be extended to three or more steps with relative errors δ_1, δ_2, and so on.

5 Numerical results

Here two model numerical examples are presented. For simplicity we assume $n_1 = n_2 = \sqrt{n}$

First is the standard 5-point Laplacian. It is a two-level Toeplitz matrix $[a_{\mathbf{i}-\mathbf{j}}]$, with free parameters a_{ij} defined as $a_{ij} = 0$, for $-n_1 + 1 \le i \le n_1 - 1$, $j = -n_2 + 1 \le j \le n_2 - 1$, except for

$$a_{00} = 4, \quad a_{0,\pm 1} = -1, \quad a_{\pm 1,0} = -1.$$

Second is a dense two-level Toeplitz matrix with a_{ij} determined by formulas

$$a_{ij} = -f(i{+}0.5, j{-}0.5){+}f(i{-}0.5, j{-}0.5){-}f(i{-}0.5, j{+}0.5){+}f(i{+}0.5, j{+}0.5),$$

where

$$f(x,y) = \frac{\sqrt{x^2+y^2}}{xy}.$$

This matrix comes from the discretization of the hypersingular integral equation [10].

The results are given in Tables 1 and 2. We calculated tensor ranks for the approximate inverse and mean displacement ranks of the factors. All computations were conducted with $\varepsilon = 10^{-5}$ (this means that "Tensor rank" and "mean displacement rank" in these tables stand for ε-ranks).

n	64^2	128^2	256^2	512^2
Running time	154 sec	333 sec	966 sec	2555 sec
Tensor rank of A^{-1}	9	10	11	12
Mean displacement rank of A^{-1}	13.5	13.5	16.8	18.6

Table 1 Numerical results for the case 1.

n	64^2	128^2	256^2	512^2
Running time	270 sec	433 sec	817 sec	1710 sec
Tensor rank of A^{-1}	13	13	12	11
Mean displacement rank of A^{-1}	8.5	9.3	9.5	9.7

Table 2 Numerical results for the case 2.

At least for these two examples we can see that the running time obeys the expected $\mathcal{O}(\sqrt{n}r_{\text{mean}}^2)$ asymptotics (where r_{mean} is a mean displacement rank; the dependence from tensor rank is hard to observe in these examples). However, we are not very satisfied with the absolute values: the constant seems to be quite large. After examining the program code it was found that the main computational efforts were spent while recompressing the results of the multiplication of two "large" (of tensor rank 5-10, approximately) TDS matrices. The multiplication using formula (10) was very fast. However, the recompression was much, much longer and it can be explained why. We have to compress the matrix of tensor rank approximately 50-100. This involves computation of many scalar products. We do not take into account that the matrix is in fact of *much lower* tensor rank (say, 10). This surely can be used in some kind of *rank-revealing* approximation of such a matrix. In the current implementation we have to calculate, in fact, the Frobenius scalar products between all the factor matrices and that is approximately 100^2 scalar products and that leads to serious slowdown. The rank-revealing version of the structured recompression will be reported elsewhere.

References

[1] D. A. Bini and B. Meini, Solving block banded block Toeplitz systems with structured blocks: algorithms and applications, *Structured Matrices: Recent Developments in Theory and Computation. Advances in Computation* (Edited by D.A.Bini, E.Tyrtyshnikov, P.Yalamov), Nova Science Publishers, Inc., Huntington, New York (2001).

[2] I. Gohberg and G. Heinig, Inversion of finite-section Toeplitz matrices consisting of elements of a non-commutative algebra, *Rev. Roum. Math. Pures et Appl.*, Vol.19, No. 5, pp. 623-663 (1974).

[3] I. Gohberg and A. A. Semencul, On inversion of finite-section Toeplitz matrices and their continuous analogues, *Matem. Issled.* (Russian), Vol.7, No. 2, pp. 201-224 (1972).

[4] W. Hackbusch, B.N. Khoromskij and E.E. Tyrtyshnikov, *Approximate iterations for structured matrices*, Max-Planck-Insitut, Leipzig, Preprint 112 (2005).

[5] G. Heinig and K. Rost, *ALgebraic methods for Toeplitz-like matrices and operators*, Berliin, Academie-Verlag (1984).

[6] H. Hotelling, Analysis of a complex of statistical variables into principal components, *J. Educ. Psych.*, 24, 1933, 417-441, 498-520.

[7] T. Kailath, S. Kung and M. Morf, Displacement ranks of matrcies and linear equations, *J. Math. Anal.and Appl.*, 68, pp. 395-407 (1979).

[8] J. Kamm and J. G. Nagy, Optimal Kronecker Product Approximations of Block Toeplitz Matrices, *SIAM J. Matrix Anal. Appl.*, Vol. 22, No. 1, pp. 155-172 (2000).

[9] V. Olshevsky, I. Oseledets, E. Tyrtyshnikov, Tensor properties of multilevel Toeplitz and related matrices, *Linear Algebra Appl.* 412 (2006), 1–21.

[10] I. Oseledets and E. Tyrtyshnikov, Approximate inversion of matrices in the process of solving a hypersingular integral equation, *Comp. Math. and Math. Phys.* **45**, No. 2 (2005), 302–313 (translated from *JVM i MF* **45**, No. 2 (2005), 315–326).

[11] V. Y. Pan, Y. Rami, Newton's iteration for the inversion of structured matrices, *Structured Matrices: Recent Developments in Theory and Computation* (Eds. Bini D.A., Tyrtyshnikov E.E., Yalamov P.), Nova Science Publishers, Huntington, New York, 2001, 79-90.

[12] G. Schulz, Iterative Berechnung der reziproken Matrix, *Z. angew. Math. und Mech.*, 13 (1), 57-59, 1933.

[13] W. F. Trench, An algorithm for the inversion of finite Toeplitz matrices, *SIAM J.Appl. Math.*, Vol. 12, pp. 515-521 (1964).

[14] E. Tyrtyshnikov, Optimal and superoptimal circulant preconditioners, *SIAM J. Matrix Anal. Appl.*, Vol. 13, No. 2, pp. 459-473 (1992).

[15] C. F. Van Loan, N. P. Pitsianis, Approximation with Kronecker products, *NATO Adv. Sci. Ser E Appl. Sci.* 232, Kluwer: Dordrecht, pp. 293–314 (1993).