Approximate displacement rank and applications

Dario Andrea Bini and Beatrice Meini

Abstract. We introduce the concepts of orthogonal displacement representation and of ϵ-displacement rank of matrix, prove some properties and show some applications to Toeplitz-like matrix inversion by means of Newton's iteration, and to cyclic reduction for the solution of block banded block Toeplitz systems.

1. Introduction and definitions

The concepts of displacement rank and displacement operator, introduced by Kailath et al. [**18, 17**], and elaborated in many other papers (see, for instance, [**15, 8, 22**] and the references cited therein), provide powerful tools for dealing with Toeplitz matrices, allow the generalization of the latter structure, and are the basis for the design of inversion formulae and of fast inversion algorithm for structured matrices. We recall that a Toeplitz matrix has entries that are constant along the diagonals.

Different displacement operators have been introduced and analyzed in the literature [**15, 8, 22, 19, 20, 21, 7**], most of them are strongly related to representation formulae for Toeplitz-like matrices and for their inverses. Operators of the kind $\Delta_S(X) = XS - SX$ defined for $X \in \mathbb{C}^{n\times n}$, where $S \in \mathbb{C}^{n\times n}$ is a given matrix, are clearly singular and their kernel is the centralizer of S. Any matrix X can be decomposed as sum of an element of the kernel and one of the image of Δ_S. Suitable choices of S provide different operators well suited for dealing with matrices of the Toeplitz type.

For instance, if $S = Z$, where $Z = (z_{i,j})$ is the $n \times n$ lower triangular Toeplitz matrix such that $z_{i+1,i} = 1$, $z_{i,j} = 0$ for $i \neq j+1$, then $\Delta_Z(X)$ is zero for any lower triangular Toeplitz matrix X, moreover, $\mathrm{rank}\Delta_Z(A) \leq 2$ for any Toeplitz matrix

1991 *Mathematics Subject Classification.* Primary ; Secondary .
Supported in part by NSF Grant #000000.

A. The following well known results hold (see for instance [**8**])

$$
\begin{aligned}
&\Delta_Z(AB) = \Delta_Z(A)B + A\Delta_Z(B),\\
&\Delta_Z(A^{-1}) = -A^{-1}\Delta_Z(A)A^{-1},\\
&A = L(A\boldsymbol{e}_1) + \textstyle\sum_{i=1}^{k} L(\boldsymbol{u}_i)L(Z\boldsymbol{v}_i)^T \text{ iff } AZ - ZA = \sum_{i=1}^{k} \boldsymbol{u}_i\boldsymbol{v}_i^T,\\
&A^{-1} = L(A^{-1}\boldsymbol{e}_1) - \textstyle\sum_{i=1}^{k} L(A^{-1}\boldsymbol{u}_i)L(ZA^{-T}\boldsymbol{v}_i)^T \text{ iff}\\
&\ AZ - ZA = \textstyle\sum_{i=1}^{k} \boldsymbol{u}_i\boldsymbol{v}_i^T,\ \det A \neq 0,
\end{aligned}
\tag{1.1}
$$

where $L(\boldsymbol{x}) = \sum_{i=1}^{n} x_i Z^{i-1}$ is the lower triangular Toeplitz matrix having $\boldsymbol{x}$ as first column, and $\boldsymbol{e}_1$ is the first column of the identity matrix. A matrix that can be written as sum of k products of lower and upper triangular Toeplitz matrices is called Toeplitz-like if k is independent of n. The representation of A given in terms of triangular Toeplitz matrices is called *displacement representation*.

We call *displacement rank* of A and denote it as $\mathrm{drk}(A)$, the rank of $\Delta_Z(A)$. In this way, a matrix with displacement rank k as well as its inverse (if nonsingular) can be written as sum of at most $k+1$ products of lower and upper triangular Toeplitz matrices.

In the original formulation of [**17, 18**] the nonsingular displacement operator $X - ZXZ^T$ has been used. This operator can be easily related to $\Delta_Z(A)$.

Similar operators and similar relations can be constructed by replacing Z with suitable matrices. It is worth citing the case $S = H = Z + Z^T$, that provides an operator suitable for Toeplitz plus Hankel matrices [**7**], or the choice $S = C = Z + \boldsymbol{e}_1\boldsymbol{e}_n^T$ (unit circulant) that is still convenient for manipulating Toeplitz matrices. Here $\boldsymbol{e}_n$ denotes the last column of the $n \times n$ identity matrix.

Although the displacement rank is kept under inversion, it is not maintained under addition and multiplication. This is an unpleasant but unavoidable situation. Consider for instance the celebrated Newton iteration

$$
\begin{aligned}
&X_{i+1} = 2X_i - X_iAX_i,\ i = 0, 1, 2, \ldots\\
&X_0 = A^T\theta,\ \theta = 1/||AA^T||
\end{aligned}
\tag{1.2}
$$

such that $\lim_i X_i = A^{-1}$, for a nonsingular matrix A. Here and hereafter $||\cdot||$ denotes any operator norm, and, if not specified, $||\cdot||$ is the Euclidean norm. If A is Toeplitz, then the displacement rank of X_0 is at most 2, but the displacement rank of X_i grows exponentially with i until the value of n is reached. This fact nullifies the computational advantages of the displacement representation so that Newton's iteration is not used in practice for Toeplitz matrix inversion for the large computational cost. In this case the displacement rank is not a useful tool for the computational treatment of the problem. An attempt to use Newton's iteration for Toeplitz inversion is done in [**23**].

Another drawback of displacement representation (and inversion) formulae is related to their numerical stability. Observe that, since the decomposition of a matrix as sum of k matrices of rank 1 is not unique, then also the displacement representation formula of (1.1) is not unique. Moreover, if some vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are very large in norm with respect to the size of the entries of A, using the displacement representation formula of (1.1) for computing a matrix-vector product surely leads to numerical cancellation problems with loss of accuracy in the computed result. Issues related to the stability of such formulae are investigated in [**14**]. Indeed, a formula constructed with vectors $\boldsymbol{u}_i$, $\boldsymbol{v}_i$ that have norms comparable with the norm of A is more desirable.

In this paper we introduce the concepts of orthogonal displacement representation and of approximate (ϵ-)displacement rank of a matrix that somehow weakens the concept of displacement rank and provides a more suitable tool for the numerical treatment of Toeplitz matrices. We show that certain numerical computations, like Newton's iteration applied to Toeplitz-like matrices and cyclic reduction for block banded block Toeplitz matrices with Toeplitz blocks [**4**], can be performed at a very low computational cost, due to the properties of the ϵ-displacement rank. Moreover, by using the orthogonal displacement representation we obtain formulae that avoid the drawback of numerical cancellation.

For notational simplicity we write $\Delta(A)$ in place of $\Delta_Z(A)$

Definition 1.1. For a given $\epsilon > 0$ define the *absolute ϵ-displacement rank* of a matrix A as

$$\mathrm{drk}_\epsilon(A) = \min_{||E|| \le \epsilon} \mathrm{rank}(\Delta(A) + E),$$

and the *relative ϵ-displacement rank* as

$$\mathrm{rdrk}_\epsilon(A) = \min_{||E|| \le \epsilon ||A||} \mathrm{rank}(\Delta(A) + E),$$

where $||\cdot||$ denotes the Euclidean norm.

If $r = \mathrm{drk}_\epsilon(A)$ ($r = \mathrm{rdrk}_\epsilon(A)$, respectively), and if $\Delta(A) = \sum_{i=1}^r \boldsymbol{u}_i \boldsymbol{v}_i^T + E$, with $||E|| \le \epsilon$ ($||E|| \le \epsilon ||A||$, respectively), from the above definition and from (1.1), we find that the matrix

$$A_\epsilon = L(A\boldsymbol{e}_1) + \sum_{i=1}^r L(\boldsymbol{u}_i) L(Z\boldsymbol{v}_i)^T$$

provides an approximation to A given in terms of the approximate displacement representation. We refer to A_ϵ as to an *ϵ-displacement (relative ϵ-displacement) representation* of A, and we call the vectors $\boldsymbol{u}_i$, $\boldsymbol{v}_i$, $i = 1, \ldots, r$, *ϵ-displacement (relative ϵ-displacement) generators* for A.

2. Main properties of ϵ-displacement rank

Let

$$\Delta(A) = U \Sigma V^T = \sum_{i=1}^k \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T, \ \sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_k > 0 \tag{2.1}$$

be the singular values decomposition (SVD) of $\Delta(A)$. We define *orthogonal displacement representation* (odr) of A (with respect to Δ) the expression

$$A = L(A\boldsymbol{e}_1) + \sum_{i=1}^k \sigma_i L(\boldsymbol{u}_i) L(Z\boldsymbol{v}_i)^T, \tag{2.2}$$

and *orthogonal displacement generator* the quadruple $(A\boldsymbol{e}_1, U, \boldsymbol{\sigma}, V)$, where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_k)$. From the uniqueness of the SVD, the odr of a matrix is unique.

Observe that, since $||\Delta(A)|| \le 2||Z|| \cdot ||A|| \le 2||A||$ and $||\Delta(A)|| = \sigma_1$, the orthogonal displacement representation of a matrix reduces the possibilities of cancellation in the computation of the matrix-vector product $A\boldsymbol{x}$.

We recall that multiplying an $n \times n$ matrix, represented through its odr (2.2) of length k, and a vector can be performed by means of $4k + 3$ FFTs and $2k + 1$ convolutions of length $2n$.

The odr of a matrix A is strictly related to the ϵ-displacement representation. The following properties provide a characterization of the ϵ-displacement rank in terms of the SVD [**4, 5**].

THEOREM 2.1. *Let* $\Delta(A) = U\Sigma V^T$ *be the singular values decomposition of* $\Delta(A)$*, where* $U, V \in \mathbb{R}^{n\times k}$*,* $U^TU = I$*,* $V^TV = I$*,* $\Sigma = \mathrm{Diag}(\sigma_1, \dots, \sigma_k)$ *and* $\sigma_1 \geq \dots \geq \sigma_k > 0$ *are the singular values of* $\Delta(A)$*. Let* $\epsilon >$ *be such that* $\epsilon < \sigma_1$*. Then* $\mathrm{drk}_\epsilon(A) = r$ *if and only if* $\sigma_r > \epsilon \geq \sigma_{r+1}$*. Moreover* $\mathrm{rdrk}_\epsilon(A) = r$ *if and only if* $\sigma_r > \epsilon||A|| \geq \sigma_{r+1}$*.*

PROOF. We prove that $\sigma_{r+1} \leq \epsilon$ if and only if $\mathrm{drk}_\epsilon(A) \leq r$. This is sufficient to prove the theorem. Let $\boldsymbol{u}_i$, $\boldsymbol{v}_i$, $i = 1, \dots, k$, be the i-th column of U and V, respectively, that is

$$\Delta(A) = \sum_{i=1}^{r} \boldsymbol{u}_i\sigma_i\boldsymbol{v}_i^T + \sum_{i=r+1}^{k} \boldsymbol{u}_i\sigma_i\boldsymbol{v}_i^T. \tag{2.3}$$

Define $E = \sum_{i=r+1}^{k} \boldsymbol{u}_i\sigma_i\boldsymbol{v}_i^T$, then $||E|| = \sigma_{r+1}$. Thus, if $\sigma_{r+1} \leq \epsilon$, we have $\mathrm{drk}_\epsilon(A) \leq \mathrm{drk}(\Delta(A) - E) = r$. Vice-versa, if $\mathrm{drk}_\epsilon(A) = r$ then there exists a matrix E, with $||E|| \leq \epsilon$, such that $\mathrm{rank}(\Delta(A) + E) = r$. Define $\theta_r = \min_{\{X:\ \mathrm{rank}(X)=r\}} ||\Delta(A) - X||$; since $\theta_r = \sigma_{r+1}$ and $\sigma_{r+1} = ||\Delta(A) - \sum_{i=1}^{r} \boldsymbol{u}_i\sigma_i\boldsymbol{v}_i^T||$, it follows that $\sigma_{r+1} = \theta_r \leq ||E||$. □

In particular, given $0 < \epsilon < \sigma_1$ and the odg $(A\boldsymbol{e}_1, U, \boldsymbol{\sigma}, V)$, of A, it is possible to compute the ϵ-displacement representation of A of lower rank in the form

$$A_\epsilon = L(A\boldsymbol{e}_1) + \sum_{i=1}^{r} \sigma_i L(\boldsymbol{u}_i) L(Z\boldsymbol{v}_i)^T$$

where r is such that $\sigma_r > \epsilon \geq \sigma_{r+1}$, or, in the case of the relative ϵ-displacement rank, $\sigma_r > \epsilon||A|| \geq \sigma_{r+1}$.

We call the above expression *approximate orthogonal displacement representation* (aodr) of A and the associated generator is called *approximate orthogonal displacement generator* (aodg).

The concept of ϵ-displacement rank has been implicitly used in [**23**] for implementing the Newton iteration for inverting Toeplitz matrices.

The following result provides a bound to the error in the ϵ-displacement representation of A.

THEOREM 2.2. *Let* $r = \mathrm{drk}_\epsilon(A) \leq \mathrm{drk}(A) = k$*, let* A_ϵ *be an* ϵ*-displacement representation of* A*, and let* $\sigma_1, \dots, \sigma_k$ *be the singular values of* $\delta(A)$*. Then it holds*

$$||A - A_\epsilon|| \leq n \sum_{i=r+1}^{k} \sigma_i \leq n(k - r)\epsilon.$$

Similarly, if $r = \mathrm{rdrk}_\epsilon(A)$ *and if* A_ϵ *is a relative* ϵ*-displacement representation of* A*, then it holds*

$$\frac{||A_\epsilon - A||}{||A||} \leq n \sum_{i=r+1}^{k} \sigma_i \leq n(k - r)\epsilon.$$

PROOF. Let $\Delta(A) = U\Sigma V^T$ be the singular values decomposition of $\Delta(A)$. From Theorem 2.1 and (2.3) it holds $A - A_\epsilon = \sum_{i=r+1}^k \sigma_i L(\boldsymbol{u}_i) L(Z\boldsymbol{v}_i)^T$, where $||\boldsymbol{u}_i|| = ||\boldsymbol{v}_i|| = 1$. Since for any vector $\boldsymbol{u}$ it holds $||L(\boldsymbol{u})|| = ||L(\boldsymbol{u})^T||$ and $||L(\boldsymbol{u})|| \le \sum_{i=1}^n |u_i| \cdot ||Z^i|| = \sum_{i=1}^n |u_i| \le \sqrt{n}||u||$, we have $||A - A_\epsilon|| \le n \sum_{i=r+1}^k \sigma_i$. The proof is completed since $|\sigma_i| \le \epsilon$ for $r = \mathrm{drk}_\epsilon(A)$ and $|\sigma_i| \le \epsilon ||A||$ for $r = \mathrm{rdrk}_\epsilon(A)$, $i = r+1, \ldots, k$. □

Conversely, we have the following

THEOREM 2.3. *If A has displacement rank k and B is an approximation of A such that $||A - B|| \le \epsilon$, then $\mathrm{drk}_{2\epsilon}(B) \le k$. Similarly, if $||B - A|| \le \epsilon ||A||$ then $\mathrm{rdrk}_{2\epsilon}(B) \le k$.*

PROOF. It holds $\Delta(B) = \Delta(A) + \Delta(B - A)$, moreover, $||\Delta(B - A)|| \le ||Z(B - A)|| + ||(B - A)Z|| \le 2||B - A|| \le 2\epsilon$ since $||Z|| = 1$. □

The following inequalities are an immediate consequence of the definition:

$$
\begin{aligned}
&\mathrm{drk}_\delta(A+B) \le \mathrm{drk}_\epsilon(A) + \mathrm{drk}_\epsilon(B), && \delta = 2\epsilon \\
&\mathrm{drk}_\delta(AB) \le \mathrm{drk}_\epsilon(A) + \mathrm{drk}_\epsilon(B), && \delta = \epsilon(||A|| + ||B||) \\
&\mathrm{drk}_\delta(A^{-1}) \le \mathrm{drk}_\epsilon(A), && \delta = ||A^{-1}||\epsilon \\
&\mathrm{rdrk}_\delta(A+B) \le \mathrm{rdrk}_\epsilon(A) + \mathrm{rdrk}_\epsilon(B), && \delta = \epsilon \tfrac{||A|| + ||B||}{||A+B||} \\
&\mathrm{rdrk}_\delta(AB) \le \mathrm{rdrk}_\epsilon(A) + \mathrm{rdrk}_\epsilon(B), && \delta = \epsilon \tfrac{||A|| \cdot ||B||}{||AB||} \\
&\mathrm{rdrk}_\delta(A^{-1}) \le \mathrm{rdrk}_\epsilon(A), && \delta = \epsilon \mu(A),
\end{aligned} \tag{2.4}
$$

where $\mu(A) = ||A|| \cdot ||A^{-1}||$ is the spectral condition number of $||A||$.

Let us introduce the operator $\mathrm{trunc}_\epsilon(\cdot)$ defined on the sets of orthogonal displacement generators in the following way:

$$
\begin{aligned}
&\mathrm{trunc}_\epsilon((\boldsymbol{a}, U, \boldsymbol{\sigma}, V)) = (\boldsymbol{a}, \widehat{U}, \widehat{\boldsymbol{\sigma}}, \widehat{V}), \\
&\widehat{\boldsymbol{\sigma}} = (\sigma_1, \ldots, \sigma_{\widehat{k}}),\ \widehat{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\widehat{k}}], \widehat{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{\widehat{k}}] \in \mathbb{R}^{n \times \widehat{k}} \\
&\sigma_{\widehat{k}} > \sigma_1 \epsilon \ge \sigma_{\widehat{k}+1},
\end{aligned} \tag{2.5}
$$

and with an abuse of notation, extend $\mathrm{trunc}_\epsilon(\cdot)$ to the set of matrices as $A_\epsilon = \mathrm{trunc}_\epsilon(A)$ where,

$$A_\epsilon = L(A\boldsymbol{e}_1) + \sum_{i=1}^{\widehat{k}} \sigma_i L(\boldsymbol{u}_i) L(ZJ\boldsymbol{v}_i)^T.$$

Clearly from Theorem 2.2 it holds

$$||A - \mathrm{trunc}_\epsilon(A)|| \le n(k - \hat{k})\sigma_1 \epsilon$$

Equation (2.5) provides a means for computing an ϵ-displacement representation of A whenever an odg is available.

REMARK 2.4. *Constructing an orthogonal (ϵ)-displacement generator.* Assume we are given $F_1, F_2 \in \mathbb{R}^{n \times k}$ such that $\Delta(A) = F_1 F_2^T$. It is almost inexpensive to compute an orthogonal displacement generator of A, i.e. the SVD of $\Delta(A)$. A cheap way to compute the SVD of $F_1 F_2^T$ consists in computing the QR factorizations $F_1 = Q_1 R_1$, and $F_2 = Q_2 R_2$, so that we have $\Delta(A) = Q_1 R_1 R_2^T Q_2^T$, and then the SVD $R_1 R_2^T = G \Sigma H^T$ of the $k \times k$ matrix $R_1 R_2^T$, where $G^T G = I$, $H^T H = I$.

Whence we obtain $\Delta(A) = U\Sigma V^T$ where $U = Q_1 G$, $V = Q_2 H$. In this way, the overall cost is $O(k^2 n)$ ops. From the odg of A it is immediate to recover an ϵ-displacement generator by means of the operator $\mathrm{trunc}_\epsilon(\cdot)$.

2.1. A more convenient displacement operator. Multiplying a Toeplitz-like matrix and a vector by means of the odr of length k involves $4k+3$ FFTs and $2k+1$ convolutions of length $2n$.

It is possible to reduce the lengths of FFTs and convolutions by a factor of 2 by means of a different displacement representation. The representation formulae that we introduce in this section are well known in the case of Toeplitz matrices [**24, 23, 16, 13, 12, 8**] and will be here derived, in their more general form, in terms of a suitable displacement operator.

Let us denote with C^+ and C^- the matrices $C^+ = Z + \boldsymbol{e}_1\boldsymbol{e}_n^T$, $C^- = Z - \boldsymbol{e}_1\boldsymbol{e}_n^T$, and set $C^+(\boldsymbol{x}) = \sum_{i=1}^n x_i (C^+)^{i-1}$, $C^-(\boldsymbol{x}) = \sum_{i=1}^n x_i (C^-)^{i-1}$ the $(+1)$-circulant and (-1)-circulant matrices having $\boldsymbol{x}$ as first column. We recall that

$$
\begin{aligned}
&C^+(\boldsymbol{x}) = F\mathrm{Diag}(\boldsymbol{y})F^H,\\
&C^-(\boldsymbol{x}) = DF\mathrm{Diag}(\hat{\boldsymbol{y}})F^H D,\\
&\boldsymbol{y} = \tfrac{1}{n}F^H\boldsymbol{x},\ \hat{\boldsymbol{y}} = \tfrac{1}{n}F^H D\boldsymbol{x},\\
&F = (\omega^{(i-1)(j-1)})_{i,j=1,\dots,n},\ \omega = \cos(2\pi/n) + \boldsymbol{i}\sin(2\pi/n)\\
&D = \mathrm{Diag}(1,\theta,\theta^2,\dots,\theta^{n-1}),\ \theta = \cos(\pi/n) + \boldsymbol{i}\sin(\pi/n),
\end{aligned}
\tag{2.6}
$$

where $\boldsymbol{i}$ is the complex unit such that $\boldsymbol{i}^2 = -1$. Introduce the invertible operators

$$
\begin{aligned}
\Delta^+(A) &= C^+A - AC^-\\
\Delta^-(A) &= C^-A - AC^+
\end{aligned}
$$

such that

$$
\begin{aligned}
\Delta^+(A) &= -\Delta(A) + \boldsymbol{e}_1\boldsymbol{e}_n^T A + A\boldsymbol{e}_1\boldsymbol{e}_n^T\\
\Delta^-(A) &= -\Delta(A) - \boldsymbol{e}_1\boldsymbol{e}_n^T A - A\boldsymbol{e}_1\boldsymbol{e}_n^T.
\end{aligned}
\tag{2.7}
$$

We have the following result

THEOREM 2.5. *It holds*

$$
\begin{aligned}
&\Delta^+(A) = \textstyle\sum_{i=1}^k \boldsymbol{u}_i\boldsymbol{v}_i^T \text{ iff } A = \frac{1}{2}\sum_{i=1}^k C^+(\boldsymbol{u}_i)C^-(J\boldsymbol{v}_i),\\
&\Delta^-(A) = \textstyle\sum_{i=1}^k \boldsymbol{u}_i\boldsymbol{v}_i^T \text{ iff } A = -\frac{1}{2}\sum_{i=1}^k C^-(\boldsymbol{u}_i)C^+(J\boldsymbol{v}_i),\\
&\Delta^-(A^{-1}) = -A^{-1}\Delta^+(A)A^{-1},\\
&\Delta^+(AB) = \Delta^+(A)B + A\Delta^+(B) - 2A\boldsymbol{e}_1\boldsymbol{e}_n^T B,\\
&\Delta^-(AB) = \Delta^-(A)B + A\Delta^-(B) + 2A\boldsymbol{e}_1\boldsymbol{e}_n^T B,
\end{aligned}
$$

where J is the permutation matrix having 1 on the anti-diagonal. In particular, if $\Delta^+(A) = \sum_{i=1}^k \boldsymbol{u}_i\boldsymbol{v}_i^T$ and $\det A \neq 0$, we obtain $A^{-1} = \frac{1}{2}\sum_{i=1}^k C^-(A^{-1}\boldsymbol{u}_i)C^+(JA^{-T}\boldsymbol{v}_i)$.

PROOF. For the linearity of the operators Δ^+ and Δ^- it is sufficient to prove the first two equations for $k = 1$, moreover, for the nonsingularity of the operators it is sufficient to prove the implications $A = \frac{1}{2}C^+(\boldsymbol{u})C^-(J\boldsymbol{v}) \Rightarrow \Delta^+(A) = \boldsymbol{u}\boldsymbol{v}^T$ and $A = \frac{1}{2}C^-(\boldsymbol{u})C^+(J\boldsymbol{v}) \Rightarrow \Delta^-(A) = \boldsymbol{u}\boldsymbol{v}^T$. For $A = \frac{1}{2}C^+(\boldsymbol{u})C^-(J\boldsymbol{v})$ we have $\Delta^+(A) = C^+A - AC^- = \frac{1}{2}(C^+(\boldsymbol{u})C^+C^-(J\boldsymbol{v}) - C^+(\boldsymbol{u})C^-C^-(J\boldsymbol{v})) = C^+(\boldsymbol{u})\boldsymbol{e}_1\boldsymbol{e}_n^T C^-(J\boldsymbol{v}) = \boldsymbol{u}\boldsymbol{v}^T$. Similarly we may do for the operator Δ^-. The remaining equations are straightforward. □

Observe that, for a Toeplitz matrix A it holds $\Delta^+(A) = \boldsymbol{e}_1(\boldsymbol{u}-\boldsymbol{v})^T J + (\boldsymbol{u}+\boldsymbol{v})\boldsymbol{e}_1^T J$, for $\boldsymbol{u} = A\boldsymbol{e}_1$, $\boldsymbol{v} = ZA\boldsymbol{e}_n$, that is, $A = \frac{1}{2}(C^+(\boldsymbol{u}+\boldsymbol{v}) + C^-(\boldsymbol{u}-\boldsymbol{v}))$, therefore

we have the following inversion formula (compare with [**24, 23, 16, 13, 12, 8**])
$A^{-1} = -\frac{1}{2}(C^{-}(A^{-1}\boldsymbol{e}_1)(I - C^{+}(\boldsymbol{w})) + (I + C^{-}(\boldsymbol{w}))C^{+}(A^{-1}\boldsymbol{e}_1)),\ \boldsymbol{w} = A^{-1}ZA\boldsymbol{e}_n.$

The new displacement representation given in the above theorem allows one to compute the product $A\boldsymbol{x}$ by means of $4k+2$ FFT's, $2k$ convolutions and $2k+1$ scalings of length n.

Indeed, we may apply the concepts of orthogonal displacement representation and of ϵ-displacement rank to the representations obtained through the new operators Δ^{+} and Δ^{-}. In this way we arrive at a more efficient implementation of this tool.

Due to the nonsingularity of Δ^{-} and Δ^{+}, the odg associated with Δ^{-} and Δ^{+}, is simply defined by the triple $(U, \boldsymbol{\sigma}, V)$ such that $U\Sigma V$ is the SVD of $\Delta^{-}(A)$ or of $\Delta^{+}(A)$, respectively, where $\Sigma = \mathrm{Diag}(\sigma_1, \dots, \sigma_k)$.

Observe also that the displacement ranks obtained by means of the operators Δ, Δ^{+} and Δ^{-} may differ at most by 2 and from the relations (2.1,2.2,2.7) we obtain the representation formula

$$\begin{aligned} A &= \tfrac{1}{2}\left(\textstyle\sum_{i=1}^{k} C^{+}(\boldsymbol{u}_i)C^{-}(J\boldsymbol{v}_i) + C^{-}(JA^T\boldsymbol{e}_n) + C^{+}(A\boldsymbol{e}_1)\right) \\ \Delta(A) &= -\textstyle\sum_{i=1}^{k} \boldsymbol{u}_i\boldsymbol{v}_i^T. \end{aligned} \tag{2.8}$$

2.2. Matrices with additional structure. Further reduction of the computational costs can be obtained if the matrix A has more structure. For instance, if A is Toeplitz, then it is symmetric with respect to the secondary diagonal (persymmetric), and it holds $A^T = JAJ$, where J is the permutation matrix having 1 on the secondary diagonal. The persymmetry of A implies that $\Delta(A)J$ is skew-symmetric.

For skew-symmetric matrices we have the following technical result:

LEMMA 2.6. *Let $A \in \mathbb{R}^{n\times n}$ be such that $A = -A^T$ and $\mathrm{rank}(A) = k$. Then $k = 2h$ is even and there exist $W \in \mathbb{R}^{n\times k}$, $D \in \mathbb{R}^{h\times h}$ such that $W^TW = I_k$, $D = \mathrm{Diag}(d_1, \dots, d_h)$ and*

$$A = W(D \otimes S)W^T,\ S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

moreover $\pm\boldsymbol{i}d_j$ are the nonnull eigenvalues of A, where $\boldsymbol{i}$ is the complex unit such that $\boldsymbol{i}^2 = -1$, and $|d_j| = \sigma_{2j-1} = \sigma_{2j}$, $j = 1, \dots, h$, where $\sigma_1, \dots, \sigma_k$ are the singular values of A.

PROOF. From the Schur normal form deduce that a skew symmetric matrix A is such that $A = QDQ^H$, for $Q^HQ = I$, $D = \mathrm{Diag}(d_j)$ where $d_{2j-1} = \theta_j\boldsymbol{i}$, $d_{2j} = -\theta_j\boldsymbol{i}$. Therefore, only an even number of eigenvalues can be nonzero. Denote $\boldsymbol{q}_j = Q\boldsymbol{e}_j$, where $\boldsymbol{e}_j$ denotes the j-th column of the identity matrix, apply the equation

$$\begin{bmatrix} \theta\boldsymbol{i} & 0 \\ 0 & -\theta\boldsymbol{i} \end{bmatrix} = \frac{\theta}{2}\begin{bmatrix} 1 & -\boldsymbol{i} \\ 1 & \boldsymbol{i} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ \boldsymbol{i} & -\boldsymbol{i} \end{bmatrix}$$

and conclude that the claim is true where the columns of W are given by $(\boldsymbol{q}_{2j-1} + \bar{\boldsymbol{q}}_{2j-1})\sqrt{2}$ and $(\boldsymbol{q}_{2j-1} - \bar{\boldsymbol{q}}_{2j-1})\boldsymbol{i}\sqrt{2}$, that are orthonormal. □

As a consequence, we have the following fundamental property:

THEOREM 2.7. [Displacement representation of a persymmetric matrix]. *If the persymmetric matrix A has displacement rank k with respect to Δ, then $k = 2h$*

and there exist matrices $W \in \mathbb{R}^{n\times k}$, $D = \mathrm{Diag}(d_1,\dots,d_h) \in \mathbb{R}^{n\times n}$ *such that* $|d_1| \ge |d_2| \ge \dots \ge |d_h|$, $W^TW = I_k$ *and*

$$\Delta(A) = W(D \otimes S)W^TJ,\ S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

Therefore,

$$A = L(\boldsymbol{a}) + \sum_{i=1}^{h} d_i(L(\boldsymbol{w}_{2i-1})L^T(ZJ\boldsymbol{w}_{2i}) - L(\boldsymbol{w}_{2i})L^T(ZJ\boldsymbol{w}_{2i-1})) \tag{2.9}$$

where $\boldsymbol{a} = A\boldsymbol{e}_1$, $\boldsymbol{w}_i = W\boldsymbol{e}_i$.

Operating with the persymmetric odr (2.9) is more convenient in terms of memory space and computational cost. For instance, the number of FFT's sufficient for computing the product of A and a vector can be reduced to $3k + 3$.

For the operators Δ^+ and Δ^- the persymmetry property provides a more structured representation that can be recovered from (2.8) where $A^T\boldsymbol{e}_n = JA\boldsymbol{e}_1$, in the light of Theorem 2.7. Also in this case the number of FFTs sufficient to compute the matrix-vector product is reduced to $3k + 3$.

3. Applications

The displacement representation allows one to compute the product between a matrix with displacement rank k and a vector in $O(kn \log n)$ operations by means of FFT. Similarly the ϵ-displacement representation allows one to approximate the matrix-vector product with the same asymptotic cost.

In this section we present two applications where sequences of matrices are generated with a displacement rank that grows exponentially while their ϵ-rank remains bounded still guaranteeing the convergence properties of the original algorithm and at the same time, reducing the complexity of the computation.

The two applications concern the inversion of Toeplitz-like matrices by means of Newton's iteration, and the solution of block banded block Toeplitz system with banded Toeplitz blocks (BBBT-BTB) with cyclic reduction.

3.1. Newton's Iteration. Newton's iteration is a valid tool for solving problems where the solution to be computed can be formally viewed as the zero of a known function. This method has been analyzed in [**8**] and, concerning Toeplitz matrices, has been investigated in [**23**] where the iteration (1.2) has been modified by replacing each X_i with the closest matrix having displacement rank 2. Condition on X_0 have been given in order that the method, modified in this way, still converges.

Here we present a different approach, based on the concept of *adaptive algorithm*, where each approximation X_i is replaced by the closest matrix having displacement rank k_i, and k_i is chosen so that the distance of A from the matrices with displacement rank k is "sufficiently" small in order to guarantee the convergence.

In this way, the displacement rank is not bounded a priori but it may vary according to the difficulty of the problem and to the needs of the computation. This approach is adaptive in the sense that the cost of the algorithm is not uniform with respect to the set of input data, where the algorithm adjusts the computation to the specific difficulty of the problem. In practice, for well conditioned matrices the growth of the ϵ-displacement rank is negligible and the overall cost is $O(n \log n)$ ops.

For (polynomially) ill-conditioned matrices the cost seems to grow as $O(n \log^2 n)$ ops. In fact, our modification of Newton's iteration requires $O(kn \log n)$ ops per step, where k is the maximum displacement rank of the matrices X_i involved in the computation. In the numerical experiments that we report at the end of this section, k takes very moderate values (say, in the range [2,6]) even for very ill-conditioned matrices.

Let A be a nonsingular matrix and consider the sequence of matrices generated, from an assigned X_0, by means of (1.2). This sequence that converges to A^{-1} if $\det A \neq 0$ for suitable X_0, is formally obtained by applying Newton's iteration $x_{i+1} = x_i - f(x_i)/f'(x_i)$ to the function $f(X) = X^{-1} - A$. Moreover we may look at (1.2) as to a particular instance of the *generalized Newton's iteration*

$$X_{i+1} = A^{-1}(I - (I - AX_i)^m), \; m \geq 2, \tag{3.1}$$

which, for $m = 2$ reduces to (1.2) and for $m = 3$ to

$$X_{i+1} = 3X_i - 3X_i AX_i + X_i(AX_i)^2. \tag{3.2}$$

3.1.1. *Convergence and regularization properties.* Denoting $R_i = I - AX_i$ the residual error at the i-th step, from (3.1) we obtain that $R_{i+1} = R_i^m$, that is,

$$R_i = R_0^{m^i}.$$

The above relation provides a lot of information on the convergence of the general iteration (3.1).

For positive definite A, with eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n = \rho(A)$, and with $X_0 = (1/\theta)I$, $\theta \geq \rho(A)$, we find that R_0 has eigenvalues $1 - \lambda_i/\theta$ between 0 and 1. Therefore $||R_i|| = (1 - \lambda_1/\theta)^{m^i}$ gives the double exponential convergence since $0 \leq 1 - \lambda_1/\theta \leq 1 - 1/\mu(A) < 1$. A simple choice for θ can be $\theta = ||A||$ for any induced norm $|| \cdot ||$.

For a general A the choice $X_0 = (1/\theta)A^T$, where $\theta \geq \rho(A^T A)$, guarantees the double exponential convergence since $R_0 = I - (1/\theta)AA^T$ is positive definite and has eigenvalues less than or equal to $1 - \mu_1/\theta$, where μ_1 is the smallest eigenvalue of AA^T.

Observe that, since $(1 - 1/x)^x < 1/e < 1/2$, after $k = \lceil \log_m(\lambda_1/\theta) \rceil$ iterations, we have $\rho(R_k) \leq 1/e < 1/2$, therefore, in order to have the residual error less than a given error bound ϵ, it is sufficient to apply the iteration (3.1) for $\nu = \lceil \log_m(\lambda_1/\theta) \rceil + \log_m \log \epsilon^{-1}$ steps, where $\mu(A) \leq \lambda_1/\theta$.

The first additive term in the above expression, that we call *overhead number of iterations*, provides the number of steps needed in order to arrive at a *pure double exponential convergence* of order m, where for pure double exponential convergence of order m we mean the error tends to zero as a^{m^i} for $a \leq 1/2$ and $m > 1$.

The second term provides the number of steps of pure double exponential convergence that occurs to reach the residual error bound ϵ.

The number of overhead iterations grows roughly as $\log \mu(A)$ and for Toeplitz matrices associate with a symbol $f(x) > 0$ the condition number is independent of n, while for $f(x) \geq 0$ the condition number grows at most polynomially with n, therefore the overhead number of iteration is $O(\log n)$.

Better choices for X_0 can be done by setting $X_0 = (1/\theta)P$, if A is positive definite, or $X_0 = (1/\theta)PP^T A^T$ where P is a preconditioner of A, i.e., a matrix such that $\mu(AP)$, is smaller than $\mu(A)$ and P has small displacement rank. There is a wide literature concerning such preconditioners for the conjugate gradient method, that can be exported to the case of Newton's iteration. For instance, the least square circulant preconditioner of [**11**] provides an effective reduction of the condition number. Moreover, the computation of P is quite easy, since $P = C^{-1}$ where the first column $\boldsymbol{c} = (c_0, \ldots, c_{n-1})$ of the circulant matrix C is given by $c_0 = a_0$, $c_i = ((n-i)a_i + ia_{n-i})/n$, $i = 1, \ldots, n-1$, where $a_0, a_1, \ldots a_{n-1}$ are the entries in the first column of A.

If P is chosen as the inverse of the banded Toeplitz matrix whose associated function is a trigonometric polynomial interpolating $f(z)$ and its derivatives at its roots, then $\mu(CA)$ is bounded from above by a constant [**9, 10**].

Newton's iteration has also very nice regularizing properties.

REMARK 3.1. *On the regularizing properties of Newton's iteration.* Consider for simplicity the case where A is positive definite. Denoting $\boldsymbol{q}_j$, $j = 1, \ldots, n$ the eigenvectors of A corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$, from the relation $A = \sum_{j=1}^n \lambda_j \boldsymbol{q}_j \boldsymbol{q}_j^T$, we deduce that $R_i = \sum_{j=1}^n (1 - \frac{\lambda_j}{\theta})^{m^i} \boldsymbol{q}_j \boldsymbol{q}_j^T$. That is, the components of the residual error along the matrices $\boldsymbol{q}_j \boldsymbol{q}_j^T$ converge to zero much faster for j close to n than for j close to 1, that is, for the eigenvectors $\boldsymbol{q}_j$ corresponding to the largest eigenvalues. Similarly we have

$$A^{-1} - X_i = \sum_{j=1}^n \frac{(1 - \lambda_j/\theta)^{m^i}}{\lambda_j} \boldsymbol{q}_j \boldsymbol{q}_j^T.$$

For very ill-conditioned matrices, X_i provides a poor approximation to A^{-1} in the eigenspace corresponding to the smallest eigenvalues and a very good approximation in the eigenspace corresponding to the largest eigenvalues even for moderate values of i. This property is very important since it provides a sort of "regularizing" behavior of Newton's iteration in the sense that a few iteration of (1.2) provide a generalized inverse of A that does not amplify the possible error in the known term of the linear system since the smallest eigenvalues of A are not yet inverted. This feature, typical of the conjugate gradient technique is called *regularization*.

3.1.2. *Using ϵ-displacement rank: adaptive Newton's iteration.* Indeed the Newton iteration for general matrices is very expensive since each step has a cost $O(n^3)$ ops versus $O(n^2)$ ops of the PCG. However, Newton's iteration may be competitive in the case of certain structured matrices where the same structure of A is shared by A^{-1}. This is the case of the displacement structure.

If for a Toeplitz (or more generally a Toeplitz-like) matrix we are able to implement a single Newton's iteration in $O(n \log n)$ steps, then we obtain an adaptive solver that require a number of ops ranging from $O(n \log n \log \log \epsilon^{-1})$ to $O(n \log^2 n + n \log n \log \log \epsilon^{-1})$ according to the conditioning of the matrix A.

Let $\mathrm{drk}(A) = k$; we will analyze the modification of Newton's iteration defined by the following recurrence

$$\begin{aligned} Y_i &= 2X_i - X_i A X_i \\ X_{i+1} &= \mathrm{trunc}_{\epsilon_i}(Y_i) \end{aligned} \tag{3.3}$$

where $\mathrm{trunc}_\epsilon(\cdot)$ is defined in (2.5) and the choice of ϵ_i can be made according to different strategies. The simplest one is to choose a constant value of ϵ_i, say, the square root of the machine precision of the computation.

A dynamic choice of ϵ that guarantees the convergence of the modified Newton method can be easily designed. Before delving in this problem observe that

$$\begin{aligned}\Delta(Y_i) &= 2\Delta(X_i) - \Delta(X_i)AX_i - X_i\Delta(A)X_i - X_iA\Delta(X_i)\\ \Delta^-(Y_i) &= 2\Delta^-(X_i) - \Delta^-(X_i)AX_i - X_i\Delta^+(A)X_i - X_iA\Delta^-(X_i)\end{aligned} \tag{3.4}$$

so that $\mathrm{drk}(Y_i) \le k + \mathrm{drk}(X_i)$ both for Δ and Δ^-, where $k = \mathrm{drk}(A)$.

Moreover, replacing for notational simplicity X_i with X and Y_i with Y, if $\Delta(X) = U_X D_X V_X^T$, or $\Delta^-(X) = U_X D_X V_X^T$, where $U_X, V_X \in \mathbb{R}^{n\times h}$ are orthogonal, and $\Delta(A) = U_A D_A V_A^T$, or $\Delta^+(A) = U_A D_A V_A^T$, where $U_A, V_A \in R^{n\times k}$ are orthogonal, we obtain that

$$\begin{aligned}&\Delta(Y) = U_Y D_Y V_Y^T\\ &U_Y = [U_X, XU_A, XAU_X],\ V_Y = [V_X, X^TU_A, X^TA^TU_X]\\ &D_Y = \begin{bmatrix} 2D_X & -D_X & 0\\ -D_X & 0 & 0\\ 0 & 0 & -D_A\end{bmatrix}\end{aligned} \tag{3.5}$$

Thus, an odr of $X_{i+1} = \mathrm{trunc}_\epsilon(Y)$ can be computed by performing the following steps:

Algorithm 1. *Computing the aodr of $Y = 2X - XAX$ (operator Δ^-).*

1. Compute the matrices U_Y, V_Y, D_Y of (3.5);
2. Compute the QR factorizations $U_Y = Q_1R_1$, $V_Y = Q_2R_2$;
3. Compute the SVD $R_1D_YR_2^T = U\Sigma V$, set $\widehat{\Sigma} = \mathrm{Diag}(\sigma_1, \dots, \sigma_r)$, where r is such that for the singular values of $R_1D_YR_2^T$ it holds $\sigma_{r+1} \le \epsilon\sigma_1 < \sigma_r$;
4. Set $U_X = Q_1\widehat{U}$, $V_X = Q_2\widehat{V}$, $D_X = \widehat{\Sigma}$, where $\widehat{U}$ and $\widehat{V}$ are made up by the first r columns of U and V, respectively.

The above algorithms can be extended to the operator Δ; in this case we need also to compute the first column of X_{i+1}, i.e., $2X_i\boldsymbol{e}_1 - X_iAX_i\boldsymbol{e}_1$ in order to complete the odg.

The cost of the above computation is about $O(hk)$ FFTs. Improvements of this computation can be obtained by suitably using further structures, like the persymmetry (as in the case of Toeplitz matrices) or the symmetry of the matrices involved, that allow the reduction of the number of FFTs. These issues, more closely related to the efficient implementation of the algorithm, are not treated in this paper.

Concerning the design of strategies for the determination of the cutting level ϵ, let us denote $D_X = (\sigma_1, \dots, \sigma_h)$, $Y = 2X - XAX$, $V = \mathrm{trunc}_\epsilon(Y) = Y + E$, $R_X = I - AX$, $R_Y = I - AY$, $R_V = I - AV$.

It clearly holds $\mathrm{drk}(Y) \le 2h + k$, $R_Y = R_X^m$, $||E|| \le 2n\sum_{i=r+1}^{2h+k}\sigma_i \le 2n(2h + k - r)\epsilon\sigma_1$ where $\sigma_r > \epsilon\sigma_1 \ge \sigma_{r+1}$, and $R_V = R_Y - AE$. Therefore $||R_V|| \le ||R_X^m|| + ||A||2n(2h + k - r)\epsilon$. Thus, we obtain that

$$||R_V|| \le ||R_X^m||(1 + \delta),\ \text{for } \epsilon \le \delta||R_X^m||/(||A||2n(2h + k)\sigma_1). \tag{3.6}$$

The above inequality is the basis to prove the following

Theorem 3.2. *Let X_0 be such that $||X_0|| \le 1 - \theta$, set $\epsilon_i = \theta/(4n(2h_i + k)\sigma_1^{(i)}||A||)$, $h_i = \mathrm{drk}(X_i)$, $\sigma_1^{(i)} = \rho(\Delta(X_i))$ ($\sigma_1^{(i)} = \rho(\Delta^-(X_i))$). Then for the residuals $R_i = I - AX_i$ of the sequence $\{X_i\}_i$ generated by the modified Newton iteration it holds $||R_i|| \le (1 - \theta/2)^{m^i}$.*

PROOF. From 3.6 it holds $||R_i|| \le ||R_{i-1}||^m(1+\theta/2)$. Whence we obtain $||R_i|| \le (1-\theta)(1+\theta/2)^{m^i} \le (1-\theta/2)^{m^i}$. □

For $\theta = 1/\mu(A)$, the bound on ϵ turns into $\epsilon \le ||R_i||^m/(4n(2h_i+k)\sigma_1\mu(A))$. Indeed, this bound guarantees the convergence whatever is the matrix A. However, in practice is too pessimistic and may lead to an unnecessary large growth of the displacement rank of the intermediate X_i. A more realistic strategy is to choose ϵ according to specific heuristics like, for instance, $\epsilon_i = \max\{||R_i||/\mu(A), \sqrt{u}\}$, where u is the machine precision of the floating point arithmetic used. The tables 1,2, 3, 4 report the values of the displacement and of the residual at the intermediate steps of the modified Newton's iteration, with no preconditioning, for different classes of matrices.

We observe that the growth of the displacement rank depends on the condition number of the matrix. However, it is much inferior to the size n of the matrix in all the test performed.

Below we resume our modification of Newton's iteration for the inversion of a Toeplitz-like matrix assigned in terms of its odr with respect to the operator Δ^-. A similar formulation can be given for the operator Δ.

ALGORITHM 2. *Newton's iteration with ϵ-displacement rank (operator Δ^-)*

INPUT: Integers n, k, the maximum number $MAXIT$ of iterations, a residual error bound ϵ, the odg $(U_A, \boldsymbol{\sigma}_A, V_A)$ of the nonsingular matrix $A \in \mathbb{R}^{n\times n}$ having displacement rank k; the odg $(U_X, \boldsymbol{\sigma}_X, \boldsymbol{V}_X)$ of a matrix X_0 such that AX_0 is positive definite and $\rho(AX_0) < 1$; a strategy for the choice of the truncation value ϵ_i.

OUTPUT: An approximation X of A^{-1} such that $||I - AX|| \le \epsilon$ given in terms of its odg $(U_X, \boldsymbol{\sigma}_X, V_X)$.

COMPUTATION:

STAGE 0: Set $i = 0$, $X = X_0$;

STAGE 1: Compute the truncation value ϵ_i;

STAGE 2: Compute the aodg $(U_Y, \boldsymbol{\sigma}_Y, V_Y)$ of $Y = 2X - XAX$, by means of algorithm 1 with $\epsilon = \epsilon_i$ and set $(U_X, \boldsymbol{\sigma}_X, V_X) = (U_Y, \boldsymbol{\sigma}_Y, V_Y)$

STAGE 3: Compute $||R||$ for the residual $R = I - AX$; if $||R|| > \epsilon$ and $i \le MAXIT$ set $i = i+1$ and continue from Stage 1.

STAGE 4: If $i > MAXIT$ then output FAILURE, otherwise output odg(X).

Concerning the choice of ϵ_i, Theorem 3.2 provides the strategy that guarantees convergence. Different strategies based on heuristics can be applied. The starting matrix X_0 can be chosen according to the features of A: for a general A, we may set $X_0 = A^T/||AA^T||$; for A symmetric and positive definite we may set $X_0 = I/||A||$. A more convenient choice can be done in terms of the least square circulant approximating A.

The computation of $||V||$, for a matrix V assigned in terms of its odg of length h, is generally expensive since it requires the computation of all the n^2 entries of V. An alternative way is to replace $||V||$ with an upper bound obtained from its odr, say, $||V||_1 \le n\sum_{i=1}^{h}\sigma_i$. This bound can be used for estimating $||A||$ as well as $||A^TA||$, and for computing $||R||$ at stage 3. The latter estimate requires the

n	$Nstep$	$Mdrk$	$Sdrk$	err
32	8	3	16	2.4e-14
64	8	3	16	5.1e-14
128	8	3	16	6.6e-14
256	8	3	16	6.8e-14
512	8	3	16	6.0e-14
1024	8	3	16	5.8e-14
2048	8	3	16	5.3e-14
4096	8	3	16	4.7e-14

TABLE 1. Matrix 1

n	$Nstep$	$Mdrk$	$Sdrk$	err
32	8	3	15	2.0e-16
64	8	3	15	4.0e-16
128	8	3	15	2.0e-16
256	8	3	15	2.0e-16
512	8	3	15	2.0e-16
1024	8	3	15	2.1e-16
2048	8	3	15	2.0e-16
4096	8	3	15	2.2e-16

TABLE 2. Matrix 2

computation of the odg of R. A cheaper way that provides a necessary but, in principle, not sufficient condition in order that $||R|| \leq \epsilon$ is $||R\boldsymbol{e}_1|| \leq \epsilon$.

3.1.3. *Numerical experiments.* In Tables 1, 2, 3, 4 we report the number of steps $Nsteps$, the maximum displacement rank $Mdrk$, the sum of the displacement ranks $Sdrk$ of each step, and the residual error err for inverting the matrix A_n, for each value of n, where $A_n = (a_{|i-j|})_{i,j=1,n}$ is defined in the following way:

Matrix 1: $a_{|i-j|} = 1/(|i-j|+1)$;
Matrix 2: $a_0 = 4, a_1 = 1, a_i = 0$ otherwise;
Matrix 3: $a_0 = 2.1, a_1 = 1, a_i = 0$ otherwise;
Matrix 4: $a_0 = 2, a_1 = 1, a_i = 0$ otherwise;

The quantity $Sdrk \cdot n \log n$ expesses the overall arithmetic cost of the algorithm.

The matrices 1,2,3 are asymptotically well conditioned while the matrix 4 has condition number that grows as $O(n^2)$.

For well conditioned matrices the number of iterations as well as the maximum displacement rank is independent of n. For the matrix 4, the number of iterations as well as the sum of the displacement ranks seems to grow as $\log_2 n$.

The number of iterations, that grows proportionally to $\log \mu(A)$, can be reduced to a constant by using a band matrix as preconditioner.

n	$Nstep$	$Mdrk$	$Sdrk$	err
32	11	4	38	3.3e-14
64	12	5	41	1.7e-15
128	11	8	52	2.0e-15
256	11	7	47	2.1e-15
512	10	6	41	3.4e-13
1024	10	5	38	2.5e-13
2048	10	5	35	1.9e-13
4096	10	4	29	2.2e-13

TABLE 3. Matrix 3

n	$Nstep$	$Mdrk$	$Sdrk$	err
32	12	5	47	3.8e-15
64	13	7	57	2.5e-15
128	14	8	68	8.9e-15
256	15	8	76	4.8e-14
512	16	10	93	6.5e-14
1024	18	13	121	1.1e-13
2048	22	19	187	3.2e-13
4096	23	22	224	1.3e-13

TABLE 4. Matrix 4

3.2. Cyclic reduction. The solution of block-tridiagonal block-Toeplitz systems

$$\begin{bmatrix} A & B & & \\ C & A & \ddots & \\ & \ddots & \ddots & B \\ & & C & A \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_n \end{bmatrix} \tag{3.7}$$

of block-size $n = 2^q+1$, can be efficiently performed by means of cyclic reduction [**1, 2, 3, 25**]. In this computation the sequence of matrices A_i, B_i, C_i, $i = 1, 2, \dots, q$, such that

$$\begin{aligned} A_{i+1} &= A_i - B_i A_i^{-1} C_i - C_i A_i^{-1} B_i \\ B_{i+1} &= -B_i A_i^{-1} B_i \\ C_{i+1} &= -C_i A_i^{-1} C_i \\ A_0 &= A, \ B_0 = B, \ C_0 = C, \end{aligned} \tag{3.8}$$

must be computed, and the block components $\boldsymbol{x}_j$, $j = 1, \dots, n$ can be easily recovered from the blocks A_i, B_i, C_i and the vectors $\boldsymbol{b}_j$.

In certain computations, for instance when the system (3.7) is obtained by reblocking a banded Toeplitz matrix [**3**] or in queuing theory problems [**6**], the blocks (3.8) have displacement rank bounded by a small constant.

When the system (3.7) is obtained by reblocking a block-banded block-Toeplitz system with banded Toeplitz blocks, the matrices (3.8) have displacement rank that may grow exponentially until it reaches the size of the blocks. This is typical in image processing problems. However, in [**4, 5**] it is observed that the approximate displacement rank of the matrices (3.8) is bounded by a small constant. Therefore a reformulation of the cyclic reduction in terms of aodr can be given.

We refer the reader to [**4**] for details on cyclic reduction where the blocks (3.8) are approximated by aodr. Here we describe how the aodg of the blocks (3.8) can be computed.

Observe that from the properties of the operators Δ and Δ^+ we have

$$\Delta(BA^{-1}B) = \Delta(B)A^{-1}B - BA^{-1}\Delta(A)A^{-1}B + BA^{-1}\Delta(B)$$
$$\Delta^+(BA^{-1}B) = \Delta^+(B)A^{-1}B - BA^{-1}\Delta^+(A)A^{-1}B + BA^{-1}\Delta^+(B).$$

Similar relations hold for $CA^{-1}C$ and $A - BA^{-1}C - CA^{-1}B$.

The above formulae can be used in the same way as (3.4,3.5) for the design of an algorithm, like Algorithm 1 for the computation of one step of approximate cyclic reduction. The computational cost of the algorithm that we obtain for the approximate solution of an $n \times n$ block-banded block-Toeplitz system with banded Toeplitz blocks, where $n = m^2$ and the blocks have size m blocks, is $O(h^2 m \log^2 m \log n + hnm \log m)$ arithmetic operations, where h is the largest ϵ-displacement rank encountered in the overall computation.

The algorithm has been implemented and tested for a problem of image restoration of $N \times N$ pixels images, with the PSF defined by the mask

$$\sigma \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \alpha & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \ \sigma = 1/(24+\alpha).$$

This mask generates a 5 block-diagonal block-Toeplitz matrix of block-size N, with blocks of size N; the off diagonal blocks are 5-diagonal Toeplitz matrices with all the nonzero entries equal to σ, the diagonal blocks are 5-diagonal Toeplitz matrices having $\sigma\alpha$ on the main diagonal and σ on the remaining nonzero diagonals. The matrix obtained in this way is reblocked into a block tridiagonal form, thus obtaining (3.7).

We have considered images of size $N = 64$, that generate a block tridiagonal matrix with block size $n = 32$, with blocks of size $m = 128$. We have reconstructed the image by solving the system with our modified CR algorithm. For $\alpha = 5$ the conditions that guarantee the boundness of the approximate displacement rank are satisfied. We have reconstructed the image with values of ϵ in the range $[10^{-6}, 10^{-3}]$, keeping a small value of the ϵ-displacement rank and an accurate precision in the approximation. Larger values of ϵ did not provide adequate approximation of the solution. The original, blurred and restored images are reported in figures 1, 2.

We have also computed the maximum values of drk_ϵ and rdrk_ϵ of the blocks generated by CR for different values of α, ϵ and the original block size N. The results are summarized in tables 5, 6, where we report the size m of the blocks of the reblocked tridiagonal matrix and the largest values of drk_ϵ and rdrk_ϵ of the blocks generated by CR, for different values of ϵ. We observe that for $\alpha = 2$ the hypotheses that guarantee the conservation of the ϵ-displacement rank are not satisfied [**4**] and even though the rdrk_ϵ depends on N, its growth is rather moderate.

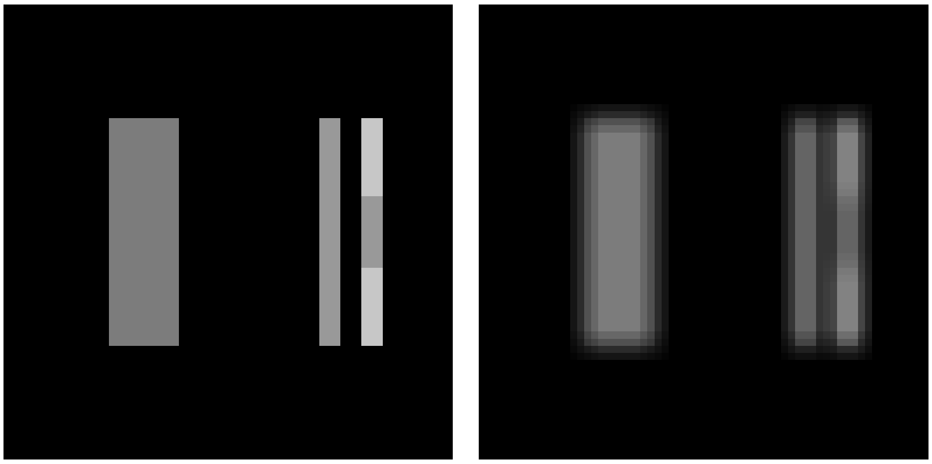

FIGURE 1. Original and blurred images

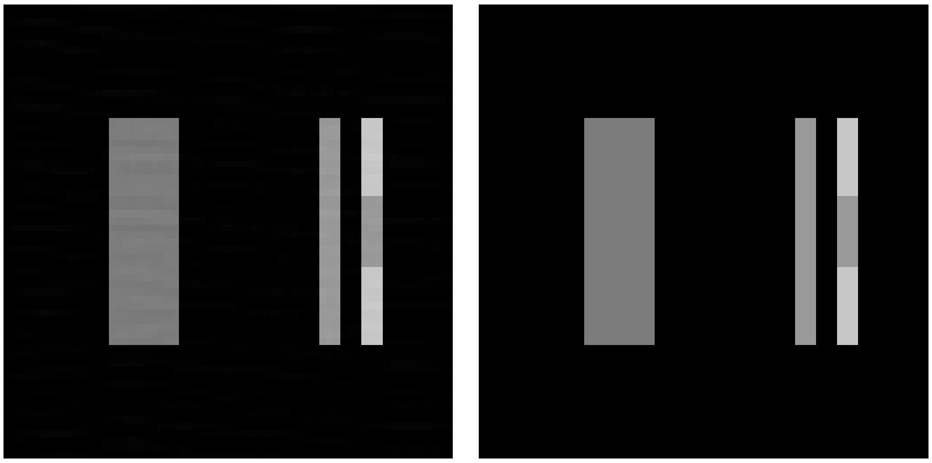

FIGURE 2. Restored images, with $\epsilon = 10^{-3}$ and $\epsilon = 10^{-6}$

	$\epsilon = 10^{-2}$		$\epsilon = 10^{-4}$		$\epsilon = 10^{-6}$	
m	drk_ϵ	rdrk_ϵ	drk_ϵ	rdrk_ϵ	drk_ϵ	rdrk_ϵ
64	12	12	20	22	28	34
128	12	12	20	22	28	34
256	12	12	20	22	28	34

TABLE 5. $\alpha = 8$

m	drk_ϵ	rdrk_ϵ
64	52	22
128	88	42
256	150	62

TABLE 6. $\alpha = 2, \epsilon = 10^{-2}$

References

1. D. A. Bini and B. Meini, *On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems*, Computations with Markov Chains (W. J. Stewart, ed.), Kluwer Academic Publisher, 1995, pp. 21–38.
2. ______, *Improved cyclic reduction for solving queueing problems*, Numerical Algorithms **15** (1997), 57–74.
3. ______, *Effective methods for solving banded Toeplitz systems*, SIAM J. Matrix Anal. Appl. **20** (1999), 700–719.
4. ______, *Solving block banded block Toeplitz systems with banded Toeplitz blocks*, Advanced Signal Processing Algorithms, Architectures, and Implementations IX (F. T. Luk, ed.), SPIE-The International Society for Optical Engineering, 1999, In press.
5. ______, *Solving block banded block Toeplitz systems with structured blocks: new algorithms and open problems*, Proceedings of WLSSC 1999, Sozopol, Bulgaria (P.Y. Yalamov S. D. Margenov, P. S. Vassilevski, ed.), 1999, In press.
6. D. A. Bini, B. Meini, and V. Ramaswami, *Analyzing M/G/1 paradigms through QBDs: the role of the block structure in computing the matrix G*, submitted for publication, 1999.
7. D. Bini and V. Pan, *Improved parallel computations with Toeplitz-like and Hankel matrices*, Linear Algebra Appl. **188–189** (1993), 3–29.
8. D. A. Bini and V. Pan, *Matrix and polynomial computations, vol. 1: Fundamental algorithms*, Birkhäuser, Boston, 1994.
9. R. H. Chan, *Toeplitz preconditioners for Toeplitz systems with nonnegative generating functions*, IMA J. of Numerical Analysis **11** (1991), 333–345.
10. R. H. Chan and M. K. Ng, *Iterative methods for linear systems with matrix structure*, Fast reliable algorithms for matrices with structure (T. Kailath and A. H. Sayed, eds.), SIAM, Philadelphia, 1999, pp. 117–154.
11. T. Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Stat. Comput. **9** (1988), 766–771.
12. I. C. Gohberg, T. Kailath, and V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp. **64** (1995), 1557–1576.
13. G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, Linear Algebra, Signal Process., IMA Vol. Math. Appl. **69** (1995), 95–114.
14. ______, *Transformation approaches for fast and stable solutions of Toeplitz systems and polynomial equations*, Proceedings of the international workshop "Recent advances in applied mathemetics", State of Kuwait, May 4–7, 1996, pp. 223–238.
15. G. Heinig and K. Rost, *Algebraic methods for Toeplitz-like matrices and operators*, Akademie-Verlag, Berlin, and Birkhäuser, Boston, 1984.
16. T. Kailath, *Displacement structure and array algorithms*, Fast reliable algorithms for matrices with structure (T. Kailath and A. H. Sayed, eds.), SIAM, Philadelphia, 1999, pp. 1–56.
17. T. Kailath, S. Kung, and M. Morf, *Displacement rank of a matrix*, Bulletin of the American Mathematical Society **1** (1979), 769–773.
18. ______, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl. **68** (1979), 395–407.
19. T. Kailath and V. Olshevsky, *Displacement structure approach to Chebyshev-Vandermonde and related matrices*, Integral Equations Operator Theory **22** (1995), 65–92.
20. ______, *Displacement structure approach to discete-trigonometric-transform based preconditioners of G. Strang type and T. Chan type*, Calcolo **33** (1996), 191–208.
21. T. Kailath and A. H. Sayed, *Displacement structure: theory and applications*, SIAM Review **37** (1995), 297–386.
22. T. Kailath and A. H. Sayed (eds.), *Fast reliable algorithms for matrices with structure*, SIAM, Philadelphia, 1999.
23. V. Y. Pan, S. Branham, R. E. Rosholt, and A.-L. Zheng, *Newton's iteration for structured matrices*, Fast reliable algorithms for matrices with structure (T. Kailath and A. H. Sayed, eds.), SIAM, Philadelphia, 1999, pp. 189–210.
24. M. Van Barel, G. Heinig, and P. Kravanja, *A stabilized superfast solver for nonsymmetric Toeplitz systems*, Tech. Report TW 293, Katholieke Universiteit Leuven, Department of Computer Science, October 1999.
25. P. Yalamov and V. Pavlov, *Stability of the block cyclic reduction*, Linear Algebra Appl. **249** (1996), 341–358.

Dipartimento di Matematica, Università di Pisa, via Buonarroti 2, I-56127 Pisa, Italy

E-mail address: bini@dm.unipi.it, meini@dm.unipi.it