Displacement structure approach to discrete-trigonometric-transform based preconditioners of G.Strang type and of T.Chan type *

Thomas Kailath and Vadim Olshevsky

Information System Laboratory,
Stanford University,
Stanford CA 94305 – 4055, U.S.A.;

E-mail : olshevsk@rascals.stanford.edu

Abstract

In this paper we use a *displacement structure* approach to design a class of new preconditioners for the *conjugate gradient method* applied to the solution of large Toeplitz linear equations. Explicit formulas are suggested for the G.Strang-type and for the T.Chan-type preconditioners belonging to any of 8 classes of matrices diagonalized by the corresponding discrete cosine or sine transforms. Under the standard Wiener class assumption the *clustering property* is established for all of these preconditioners, guaranteeing a rapid convergence of the preconditioned conjugate gradient method. All the computations related to the new preconditioners can be done in real arithmetic, and to fully exploit this advantageous property one has to suggest a fast real-arithmetic algorithm for multiplication of a Toeplitz matrix by a vector. It turns out that the obtained formulas for the G.Strang-type preconditioners allow a number of representations for Toeplitz matrices leading to a wide variety of real-arithmetic multiplication algorithms based on any of 8 discrete cosine or sine transforms.

Recently transformations of Toeplitz matrices to Vandermonde-like or Cauchy-like matrices have been found to be useful in developing accurate *direct* methods for Toeplitz linear equations. In this paper we suggest to further extend the range of the transformation approach by exploring it for *iterative* methods; this technique allowed us to reduce the complexity of each iteration of the preconditioned conjugate gradient method.

We conclude the paper with a suggestion on how to exploit the displacement structure to efficiently organize numerical experiments in a numerically reliable way.

1 Introduction

1.1. PCGM for Toeplitz linear equations. We consider the solution of a large linear system of equations $A_m x = b$ whose coefficient matrix A_m is a $m \times m$ leading submatrix of a single-infinite

*This work was supported in part by ARO contract DAAH04-96-1-0176, NSF contract CCR-962811 and also by DARPA contract F49620-95-1-0525. The views and conclusions contained in these documents are those of the author(s) and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Science Foundation, the Advanced Research Projects Agency, the Army Research office, or the U.S. Government.

symmetric Toeplitz matrix of the form

$$A = \left[\ a_{|i-j|}\ \right] = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots \\ a_1 & a_0 & a_1 & a_2 & \ddots \\ a_2 & a_1 & a_0 & a_1 & \ddots \\ a_3 & a_2 & a_1 & a_0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{1.1}$$

usually associated with the corresponding *generating function* $a(z) = \sum_{k=-\infty}^{\infty} a_{|k|} z^k$. Gaussian elimination ignores any special structure, thus requiring $O(m^3)$ arithmetic operations to solve $A_m x = b$. There is a number of *fast* Toeplitz solvers all taking advantage of the structure (1.1) to significantly reduce the number of operations. For example, the classical Schur and Levinson algorithms, see, e.g., [K87], and references therein, each requires only $O(m^2)$ operations per system. Moreover the there are even *super-fast* Toeplitz solvers with a smaller complexity of $O(m \log^2 m)$ operations. The numerical stability of such *direct* Toeplitz solvers is discussed in a number of resent papers, revealing that although fast algorithms were believed to be unstable by their very nature, there are methods to obtain simultaneously fast and accurate solution.

Along with considerable current efforts to develop and to stabilize direct methods, the preconditioned conjugate gradient method (PCGM) for solving Toeplitz linear systems has garnered much attention. This is a well-known iterative procedure which computes at each iteration step two inner products of length m and one multiplication of the coefficient matrix by a vector, thus requiring $O(m \log m)$ operations per iteration. The number of iterations depends upon the clustering of the spectrum of the A_m, and if the latter has $m - s$ eigenvalues clustered around 1, then PCGM will converge in only s iterations, see, e.g., [GL89].

Classical results on the eigenvalue distribution of Toeplitz matrices (see, e.g., [GS84]) indicate that we cannot expect, in general, any clustering, and the convergence of the method will be slow. This disadvantage motivated G.Strang to propose the use of a certain *circulant* matrix P to reduce the number of iterations. The idea was to apply the algorithm to a *preconditioned system*

$$P^{-1}Ax = P^{-1}b.$$

where the *preconditioner* P should satisfy the following three requirements.

Property 1. *The complexity of the construction of P should be small, not exceeding $O(m \log m)$ operations.*

Property 2. *A linear system with P should be solved in $O(m \log m)$ operations.*

Property 3. *The spectrum of $P^{-1}A_m$ should be clustered around 1, more precisely the following holds :*

- *For any $\varepsilon > 0$ there exist integers N and s such that for any $m > N$, at most s eigenvalues of $P^{-1}A$ lie outside the interval* $[1 - \varepsilon, 1 + \varepsilon]$.

Summarizing, if a preconditioner satisfying the above properties 1-3 can be constructed, then the complexity of the PCGM will be reduced to only $O(m \log m)$ operations, which will be even less than the complexity of superfast direct methods.

The first (now well-known) proposed preconditioners of G.Strang [S86] and of T.Chan [C88] were *circulant* matrices, defined respectively by

$$S(A_m) = \text{ circ } (a_0, a_1, a_2, \ldots, a_2, a_1),$$

$$C(A_m) = \text{ circ } (a_0,\ \frac{m-1}{m}a_1 + \frac{1}{m}a_{m-1},\ \frac{m-2}{m}a_2 + \frac{2}{m}a_{m-2},\ \ldots,\ \frac{1}{m}a_{m-1} + \frac{m-1}{m}a_1).$$

Here circ (r) denotes a circulant matrix specified by its first row r. For these two preconditioners the first property holds by their construction, and since circulant matrices are diagonalized by the discrete Fourier transform (DFT) matrix $\mathcal{F}$, the second property is also immediately satisfied. Moreover for the case when the generating function $a(z) = \sum_{k=-\infty}^{\infty} a_{|k|} z^k$ is a function from the Wiener class, positive on the unit circle, the 3rd property for the G.Strang and T.Chan preconditioners was established in [C89], [CS89] and in [CY92], resp.

A recent survey [CN96] gives a fairly comprehensive review of these and related results, and describes many other preconditioners, including those of R.Chan, E.Tyrtyshnikov, T.Ku and C.Kuo, T.Huckle, and others. (A thorough theoretical and numerical comparison of all different preconditioners is one of the directions of current research, indicating that the question of "which preconditioner is better" may have different answers depending upon the particular classes of Toeplitz systems, and their generating functions, see, e.g., [TS96], [T95], [CN96].)

Along with many favorable properties of circulant preconditioners, they unfortunately require complex arithmetic (for computing FFT's), even for real symmetric Toeplitz matrices. To overcome this disadvantage, D.Bini and F.Di Benedetto [BB90] proposed *non-circulant* analogs of the G.Strang and of T. Chan preconditioners, belonging to the so-called τ-class (introduced in [BC83] as the class of all matrices diagonalized by the (*real*) discrete sine I transform (DST-I) matrix). D.Bini and F.Di Benedetto established for their preconditioners the properties 1-3 under the Wiener class assumption.

In this paper we continue the work started in [S86], [C88], [BB90], and give a systematic account of G.Strang-type and of T.Chan-type preconditioners belonging to the classes of matrices diagonalized by other real trigonometric transforms (we consider 4 discrete cosine and 4 discrete sine transforms). For each of these 8 cases we derive explicit formulas for the G.Strang-type and the T.Chan-type preconditioners and establish for them the above properties 1-3 (under the standard Wiener class assumption).

This problem, perhaps, could be solved directly, but we have found that an interpretation in terms of *displacement structure* [KKM79] and of *partially reconstructible* matrices [KO95a] often allows us to simplify many arguments. We believe that the displacement structure approach (systematically exposed in this contribution) will be useful in addressing other problems related to preconditioning, and a recent work [CNP94], [H95] supports this anticipation.

1.2. Displacement structure approach. We next use the results of [KO95a] to briefly give an interpretation of the classical G.Strang and T.Chan circulant preconditioners in terms of *partially reconstructible matrices*. This technique will be further extended in the main text below. The displacement structure approach initiated by [KKM79] is based on introducing in a linear space of all $m \times m$ matrices a suitable displacement operator $\nabla(\cdot) : \mathbf{R}^{m\times m} \to \mathbf{R}^{m\times m}$ of the form

$$\nabla(R) = R - FRF^T, \qquad \text{or} \qquad \nabla(R) = F^T R - RF. \tag{1.2}$$

A matrix R is said to have ∇-displacement structure, if it is mapped to a low-rank matrix $\nabla(R)$. Since a low-rank matrix can be described by a small number of parameters, a representation of a matrix by its image $\nabla(R)$ often leads to interesting results, and is useful for the design of many fast algorithms. This approach has been found to be useful for studying many different patterns of structure (for example, Toeplitz, Vandermonde, Cauchy, etc.) by specifying for each of them an appropriate displacement operator. For example, *Toeplitz-like matrices* are defined as having displacement structure with respect to the choice

$$\nabla_{Z_1}(R) = R - Z_1 R Z_1^T, \tag{1.3}$$

where $Z_1 =$ circ $(0, \ldots, 0, 1)$. The motivation for the above definition can be inferred from the easily verified fact that for any Toeplitz matrix A the rank of $\nabla_{Z_1}(A)$ does not exceed 2. Although the latter definition of Toeplitz-like matrices was used by several authors, it is slightly different from the standard one,

$$\nabla_{Z_0}(R) = R - Z_0 R Z_0^T$$

where Z_0 is the lower shift matrix. The crucial difference is that ∇_{Z_1} clearly has a *nontrivial kernel*, so the image $\nabla_{Z_1}(R)$ no longer contains all the information on R. Such matrices R have been called *partially reconstructible* in [KO95a], and systematically studied there. In the Toeplitz-like case Ker ∇_{Z_1} coincides with the subspace of all circulant matrices in $\mathbf{R}^{m \times m}$, so we can observe that the G.Strang and T.Chan preconditioners are both chosen from Ker ∇_{Z_1}.

1.3. A proposal : ∇_{H_Q}-kernel preconditioner. The above displacement operator ∇_{Z_1} is not the only one associated with the class of Toeplitz matrices. We propose to apply the above interpretation, and develop the analogs of G.Strang and T.Chan preconditioners in the kernels of several other related displacement operators of the form

$$\nabla_{H_Q}(R) = H_Q^T R - R H_Q. \tag{1.4}$$

Moreover, we shall specify 8 matrices H_Q for which the kernel of the corresponding displacement operator (1.4) coincides with the subspace of matrices diagonalized by any one of the 8 known versions of discrete cosine/sine transforms. For each of these cases we write down the formulas for the corresponding G.Strang-type preconditioner and T.Chan-type preconditioner. Under the standard Wiener class assumption we establish for these new preconditioners the properties 1-3.

1.4. Fast real-arithmetic multiplication of a Toeplitz matrix by a vector. As was mentioned above, each iteration of the PCGM involves a multiplication of the preconditioned matrix $P^{-1}A$ by a vector. All the computations related to the new preconditioners can be done in real arithmetic. However, the standard technique for the multiplication of a Toeplitz matrix by a vector is based on the FFT, thus requiring complex arithmetic. We show that in each of the considered cases the new formulas for the G.Strang-type preconditioners allow us an embedding of a $m \times m$ matrix A into a larger $2m \times 2m$ matrix, which is diagonalized by the corresponding (real) discrete cosine/sine transform matrix. This observation allows us to suggest a variety of new $O(m \log m)$ real-arithmetic algorithms for the multiplication of a Toeplitz matrix by a vector, using any of 8 versions of discrete cosine or sine transforms. For the DST-I case such an algorithm was suggested earlier in [BK95].

1.5. Transformations. Toeplitz-like matrices display just one kind of displacement structure and the following two displacement operators

$$\nabla(R) = R - D R Z_1^T, \qquad \nabla(R) = R - DRD$$

(with a diagonal D) are used to define the classes of Vandermonde-like and Cauchy-like matrices, respectively. It was observed in [P90] and in [GO94a], [He95a] that matrices with displacement structure can be transformed from one class to another. In particular (cf. [GO94a]) the fact that $Z_1 = \mathcal{F}^* D \mathcal{F}$ where $\mathcal{F}$ is the normalized DFT matrix, allows us to transform a Toeplitz-like matrix R into a Vandermonde-like matrix $\mathcal{F}A$ and a Cauchy-like matrix $\mathcal{F}A\mathcal{F}^*$. Since Vandermonde-like and Cauchy-like matrices allow introducing pivoting into fast Gaussian elimination algorithms (cf. [GO94b], Alg. 7.1 (partial pivoting) and Alg. 6.1 (symmetric pivoting)), this idea has been found to be useful to numerically reliable *direct* methods for solving Toeplitz linear equations, see, e.g., [He95a]; for the first accurate algorithms of this kind see [GKO95], [KO95a] , as well as [KO95b], [KO94], [BKO94], [SB95], [Gu95], [Gu96].

In this paper we suggest to exploit this technique for *iterative* methods, and to replace a preconditioned system $P^{-1}Ax = P^{-1}b$ by a transformed system $(\mathcal{F}P^{-1}\mathcal{F}^*)(\mathcal{F}A)x = \mathcal{F}P^{-1}b$. An advantage for circulant preconditioners is that the transformed preconditioner $\mathcal{F}P\mathcal{F}^*$ is a diagonal matrix, which allows us to ave 2 FFT's at each iteration (a Vandermonde-like matrix $\mathcal{F}A$ can be multiplied by a vector with exactly the same complexity as for the initial Toeplitz matrix A) [1]. We exploit the fact that any of 8 discrete cosine/sine transforms can transform a Toeplitz-like matrix to a Vandermonde-like matrix. Therefore each of the new discrete-transform preconditioners for a Toeplitz matrix A is transformed to a diagonal preconditioner for the corresponding Vandermonde-like matrix (and Cauchy-like) matrices. Moreover, for each of these cases we propose new real-arithmetic algorithms for multiplication of the corresponding Vandermonde-like matrix by a vector, with the same complexity as for the initial Toeplitz matrix (this technique also suggests another set of new real-arithmetic algorithms for multiplication of a Toeplitz matrix by a vector). Therefore in each of the 8 considered cases a preconditioned system can be replaced by a transformed system in which new preconditioners became just diagonal matrices, thus allowing us to reduce complexity of one iteration to just 4 real discrete cosine/sine transforms of the order m.

Of course, more work need to be done to theoretically and numerically compare the new preconditioners with the many existing ones, and we conclude the paper with a suggestion on how to exploit the displacement structure to efficiently organize experiments in a numerically reliable manner.

2 Partially reconstructible matrices

We shall address the problem of constructing discrete-transform based preconditioners in the second part of the paper, and start here with necessary definitions and related facts on displacement structure and partially reconstructible matrices. Let us consider a displacement operator

$$\nabla_{\{F,A\}}(R) = F \cdot R - R \cdot A. \tag{2.1}$$

and recall the following standard definitions.

- A number $\alpha = \text{rank}\nabla_{\{F,A\}}(R)$ is called the $\nabla_{\{F,A\}}$-*displacement rank* of R. (A matrix R is said to have a $\nabla_{\{F,A\}}$-*displacement structure* if α is small compared to the size.)
- A pair of rectangular $n \times \alpha$ matrices $\{G, B\}$ in any possible factorization

$$\nabla_{\{F,A\}}(R) = F \cdot R - R \cdot A = G \cdot B^T, \tag{2.2}$$

  is called a $\nabla_{\{F,A\}}$-*generator* of R;

If the matrices F and A have no common eigenvalues, $\nabla_{\{F,A\}}$ is invertible, so its generator contains a complete information on R. For our purposes in this paper it will be necessary to consider another case where the displacement operator

$$\nabla_F(R) = F^T \cdot R - R \cdot F \tag{2.3}$$

clearly has a nontrivial kernel. Such R have been called *partially reconstructible* in [KO95a], because now only part of the information on R is contained in $\{G, B\}$. Following [KO95a] we shall refer to a triple $\{G, J, R_{\mathcal{K}}\}$ as a ∇_F-generator of R, where the latter three matrices are defined as follows.

[1] I.e., using a decomposition of A into a sum of a circulant and skew-circulant matrices, multiplied by $\mathcal{F}$.

- Since $\nabla_F(R^T) = -\nabla_F(R)$, we can write

$$\nabla_F(R) = F^T \cdot R - R \cdot F = G \cdot J \cdot G^T, \qquad \text{with} \qquad J^T = -J \in \mathbf{R}^{\alpha \times \alpha} \tag{2.4}$$

- Further, let us decompose

$$R = R_{\mathcal{K}} + R_{\mathcal{K}^\perp} \qquad \text{with respect to} \qquad \mathbf{R}^{n \times n} = \mathcal{K} \oplus \mathcal{K}^\perp. \tag{2.5}$$

where $\mathcal{K} =$ Ker ∇_F and the orthogonality in $\mathbf{R}^{n \times n}$ is defined using the inner product

$$< A, B >= \mathrm{tr}(B^* \cdot A), \qquad A, B \in \mathbf{R}^{n \times n}, \tag{2.6}$$

with $\mathrm{tr}(A)$ denoting the sum of all diagonal entries of A, or, equivalently, the sum of eigenvalues of A. Note that the latter inner product induces the Frobenius norm in $\mathbf{R}^{n \times n}$.

Clearly, now all the information on R is contained in the newly defined generator, $\{G, J, R_{\mathcal{K}}\}$.

3 Polynomial Hankel-like matrices

3.1. Polynomial Hankel-like matrices. In this paper we exploit a special displacement operator

$$\nabla_{H_Q}(R) = H_Q^T \cdot R - R \cdot H_Q = GJG^T, \tag{3.1}$$

with the upper *Hessenberg* matrix

$$H_Q = \begin{bmatrix} a_{01} & a_{02} & \cdots & & \cdots & a_{0,n} \\ a_{11} & a_{12} & \cdots & & \cdots & a_{1,n} \\ 0 & a_{22} & \cdots & & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & & a_{n-1,n-1} & a_{n-1,n} \end{bmatrix}. \tag{3.2}$$

The latter has been called in [MB79] a *confederate* matrix of the associated system of polynomials $Q = \{Q_0(x), Q_1(x), ..., Q_n(x)\}$ defined by

$$x \cdot Q_{k-1}(x) = a_{k,k} \cdot Q_k(x) + a_{k-1,k} \cdot Q_{k-1}(x) + ... + a_{0,k} \cdot Q_0(x). \tag{3.3}$$

We shall refer to matrices having low ∇_{H_Q}-displacement rank as *polynomial Hankel-like matrices*, an explanation for this nomenclature will be offered in Sec. 3.4 after presenting the following example.

3.2. Example. Classical Hankel and Hankel-like matrices. For the simplest polynomial system $P = \{1, x, x^2, \ldots, x^{n-1}, Q_n(x)\}$, its confederate matrix trivially reduces to the companion matrix

$$H_P = \begin{bmatrix} 0 & 0 & \cdots & 0 & -\frac{q_0}{q_n} \\ 1 & 0 & \cdots & 0 & -\frac{q_1}{q_n} \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -\frac{q_{n-1}}{q_n} \end{bmatrix},$$

of the $Q_n(x) = q_n x^n + \ldots + q_1 x + q_0$. Now it is immediate to check that the shift-invariance-property of a Hankel matrix, $R = \left[\begin{array}{c} h_{i+j} \end{array}\right]_{0\le i,j\le n-1}$ implies that

$$\nabla_{H_P}(R) = H_P^T R - R H_P = \left[\begin{array}{cc} e_n & g \end{array}\right] \left[\begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array}\right] \left[\begin{array}{cc} e_n & g \end{array}\right]^T, \tag{3.4}$$

where e_n is the last coordinate vector and

$$g = \left[\begin{array}{c} h_{n-2} \\ \vdots \\ h_{2n-2} \\ 0 \end{array}\right] + \frac{1}{q_n} H \left[\begin{array}{c} q_0 \\ \vdots \\ q_{n-2} \\ q_{n-1} \end{array}\right].$$

Briefly, the ∇_{H_P}-displacement rank of an arbitrary Hankel matrix does not exceed two. Hence matrices with small ∇_{H_P}-displacement rank (not just $\alpha \le 2$) are referred to as *Hankel-like* matrices.

3.3. Diagonalization of confederate matrices. To explain the name polynomial Hankel-like matrices we shall need the following result, which will be widely used in what follows. It can be easily checked by direct multiplication (cf. with [MB79]) that the confederate matrix is diagonalized by the *polynomial Vandermonde matrix* V_Q,

$$H_Q = V_Q^{-1} D_x V_Q, \tag{3.5}$$

where

$$V_Q = \left[\begin{array}{cccc} Q_0(x_1) & Q_1(x_1) & \cdots & Q_{n-1}(x_1) \\ Q_0(x_2) & Q_1(x_2) & \cdots & Q_{n-1}(x_2) \\ \vdots & \vdots & & \vdots \\ Q_0(x_n) & Q_1(x_n) & \cdots & Q_{n-1}(x_n) \end{array}\right] \qquad D_x = \operatorname{diag}(x_1, x_2, ..., x_n). \tag{3.6}$$

Here $\{x_k\}$ are the zeros of $Q(_n(x)$, which in our application here always will be n distinct numbers, so we shall impose this restriction throughout the paper.

3.4. Change of basis. Since $H_P = V_P^{-1} D_x V_P$ (see, e.g., (3.5)), we have

$$H_Q = S_{PQ}^{-1} H_P S_{PQ}, \qquad \text{with} \qquad S_{PQ} = V_P^{-1} V_Q. \tag{3.7}$$

Now using (3.7) and (3.4) one sees that the ∇_{H_Q}-displacement rank of $S_{PQ}^T R S_{PQ}$ (with Hankel R) also does not exceed 2. We refer to such $S_{PQ}^T R S_{PQ}$ as *polynomial Hankel* matrices (or Hankel matrices represented in the polynomial basis Q), because the similarity matrix $S_{PQ} = \left[\begin{array}{c} s_{i,j} \end{array}\right]_{1\le i,j\le n}$ can be easily shown to be an upper triangular matrix with the entries being the coefficients of $Q_k(x) = \sum_{i=0}^{k} s_{i+1,k+1} x^k$. Therefore the more general matrices having low ∇_{H_Q}-displacement rank (not just $\alpha \le 2$) are called *polynomial Hankel-like* matrices.

Since (3.1) has a nontrivial kernel, such R are partially reconstructible. Our next goal is to describe the kernel of ∇_{H_Q}.

4 Transformation to Cauchy-like matrices and the kernel of ∇_{H_Q}

Recently transformation of structured matrices from one class to another has been found to be useful to design for them many efficient algorithms. In this paper we exploit an approach of [KO95a] for transformation of partially reconstructible matrices to transform polynomial Hankel-like matrices into Cauchy-like matrices, defined as having low displacement rank with respect to the simplest displacement operator

$$\nabla_{D_x}(R) = D_x R - R D_x \tag{4.1}$$

with a diagonal matrix D_x. In fact, (3.5) immediately implies the following statement.

Proposition 4.1 *Let R be a polynomial Hankel-like matrix in (3.1), given by its ∇_{H_Q}-generator $\{G, J, R_{\mathcal{K}}\}$, and W_Q denotes an arbitrary invertible diagonal matrix. Then $W_Q^{-T} V_Q^{-T} R V_Q^{-1} W_Q^{-1}$ is a Cauchy-like matrix with a ∇_{D_x}-generator*

$$\{W_Q^{-T} V_Q^{-T} G, \;\; J, \;\; W_Q^{-T} V_Q^{-T} R_{\mathcal{K}} V_Q^{-1} W_Q^{-1}\}. \tag{4.2}$$

Since the kernel of ∇_{D_x} is easy to describe (it is the subspace of all diagonal matrices), the above proposition implies the next statement.

Proposition 4.2 *Let H_Q, V_Q and D_x be defined by (3.2) and (3.6), where we assume that the diagonal entries of D_x are m different numbers. The kernel of $\nabla_{H_Q}(\cdot)$ in (3.1) has the form*

$$\mathcal{K} = \text{span}\{(H_Q^T)^k \cdot (V_Q^T W_Q^2 V_Q), \qquad k = 0, 1, \ldots n-1\}. \tag{4.3}$$

where W_Q is an arbitrary invertible diagonal matrix.

Finally, by replacing in (4.3) powers $(H_Q^T)^k$ by $Q_k(H_Q^T)$, and using (4.2) we obtain the following statement.

Corollary 4.3 *A matrix $R \in \mathcal{K} =$ Ker ∇_{H_Q} given by*

$$R = \sum_{k=0}^{n-1} r_k \cdot Q_k(H_Q^T) \cdot (V_Q^T W_Q^2 V_Q),$$

can be diagonalized as follows :

$$W_Q^{-T} V_Q^{-T} R V_Q^{-1} W_Q^{-1} = \begin{bmatrix} r(x_1) & & \\ & \ddots & \\ & & r(x_n) \end{bmatrix}$$

where the diagonal entries are computed via a polynomial Vandermonde transform

$$\begin{bmatrix} r(x_1) \\ \vdots \\ r(x_n) \end{bmatrix} = V_Q \begin{bmatrix} r_0 \\ \vdots \\ r_{n-1} \end{bmatrix}$$

Here we may note that the idea of displacement is to replace operations on the n^2 entries of a $n \times n$ structured matrix by manipulation on a smaller number $O(n)$ of parameters. The results of Sec. 3 and 4 are based on the displacement equation (3.1), which describes R by the entries of H_Q and $\{G, J, R_{\mathcal{K}}\}$. In the general situation matrix H_Q itself involves $O(n^2)$ parameters, so such a representation is no longer efficient. In the next section we specialize the results of Sec. 3 and 4, and list 8 cases for which the above approach is beneficial.

5 Orthonormal polynomials and discrete trigonometric transforms

5.1. Orthonormal polynomials. Examination of the propositions in the previous section indicate that the kernel of ∇_{H_Q} will have the simplest form in the case when there is a diagonal matrix W_Q such that the matrix $T_Q = W_Q V_Q$ is orthonormal, see, e.g., (4.3). It is easy to see that the latter condition is satisfied when the polynomials in $\{Q_k(x)\}$ are orthonormal with respect to the discrete inner product

$$< p(x), q(x) >= \sum_{k=1}^{n} p(x_k) q(x_k) w_k^2,$$

where the nodes $\{x_k\}$ are the zeros of $Q_n(x)$, and the weights w_k are diagonal entries of W_Q. Moreover, in this case the polynomials $\{Q_k(x)\}$ satisfy three-term recurrence relations so their confederate matrix reduces to the corresponding Jacobi (i.e., symmetric tridiagonal) matrix.

5.2. Discrete cosine and sine transforms. Recall that our aim in this paper is to construct preconditioners diagonalized by discrete cosine or sine transform matrices, formally defined in the next table, where

$$\eta_k = \begin{cases} \frac{1}{\sqrt{2}} & k = 0, N \\ 1 & \text{otherwise} \end{cases} .$$

Table 1. Discrete trigonometric transforms.

		Discrete transform	Inverse transform
DCT-I	$C_N^I =$	$\sqrt{\frac{2}{N-1}} \left[\eta_k \eta_{N-1-k} \eta_j \eta_{N-1-j} \cos \frac{kj\pi}{N-1} \right]_{k,j=0}^{N-1}$	$[C_N^I]^{-1} = [C_N^I]^T = C_N^I$
DCT-II	$C_N^{II} =$	$\sqrt{\frac{2}{N}} \left[\eta_k \cos \frac{k(2j+1)\pi}{2N} \right]_{k,j=0}^{N-1}$	$[C_N^{II}]^{-1} = [C_N^{II}]^T = C_N^{III}$
DCT-III	$C_N^{III} =$	$\sqrt{\frac{2}{N}} \left[\eta_j \cos \frac{(2k+1)j\pi}{2N} \right]_{k,j=0}^{N-1}$	$[C_N^{III}]^{-1} = [C_N^{III}]^T = C_N^{II}$
DCT-IV	$C_N^{IV} =$	$\sqrt{\frac{2}{N}} \left[\cos \frac{(2k+1)(2j+1)\pi}{4N} \right]_{k,j=0}^{N-1}$	$[C_N^{IV}]^{-1} = [C_N^{IV}]^T = C_N^{IV}$
DST-I	$S_N^I =$	$\sqrt{\frac{2}{N+1}} \left[\sin \frac{kj}{N+1}\pi \right]_{k,j=1}^{N}$	$[S_N^I]^{-1} = [S_N^I]^T = S_N^I$
DST-II	$S_N^{II} =$	$\sqrt{\frac{2}{N}} \left[\eta_k \sin \frac{k(2j-1)}{2N}\pi \right]_{k,j=1}^{N}$	$[S_N^{II}]^{-1} = [S_N^{II}]^T = S_N^{III}$
DST-III	$S_N^{III} =$	$\sqrt{\frac{2}{N}} \left[\eta_j \sin \frac{(2k-1)j}{2N}\pi \right]_{k,j=1}^{N}$	$[S_N^{III}]^{-1} = [S_N^{III}]^T = S_N^{II}$
DST-IV	$S_N^{IV} =$	$\sqrt{\frac{2}{N}} \left[\sin \frac{(2k-1)(2j-1)}{4N}\pi \right]_{k,j=1}^{N}$	$[S_N^{IV}]^{-1} = [S_N^{IV}]^T = S_N^{IV}$

Note that we use a slightly different definition for the DCT-I, ensuring that now all the discrete transform matrices in Table 1 are orthogonal. The modified DCT-I can be transformed into the regular one by just appropriate scaling.

Now using the fact that the Chebyshev polynomials of the first and second kind,

$$T_k(x) = \cos(k \arccos x), \qquad U_k = \frac{\sin((k+1) \arccos x)}{\sin(\arccos x)} \tag{5.1}$$

are essentially cosines and sines, we obtain that all the discrete transform matrices T_Q in Table 1 can be seen as the orthogonal matrices $W_Q \cdot V_Q$ defined by orthonormal (Chebyshev-like) polynomials $\{Q_k(x)\}_{k=0}^{n-1}\}$, with the weight matrices W_Q specified in Tables 2 and 3. We therefore adopt the designation

$$T_Q = W_Q V_Q$$

for all eight transform matrices in Table 1 by associated them with the corresponding polynomial systems Q.

Table 2. First n polynomials.

	$\{Q_0,$	Q_1	$\ldots,$	Q_{n-2}	$Q_{n-1}\}$
DCT-I	$\{\frac{1}{\sqrt{2}}T_0,$	$T_1,$	$\ldots,$	$T_{n-2},$	$\frac{1}{\sqrt{2}}T_{n-1}\}$
DCT-II	$\{U_0,$	$U_1-U_0,$	$\ldots,$		$U_{n-1}-U_{n-2}\}$
DCT-III	$\{\frac{1}{\sqrt{2}}T_0,$	$T_1,$	$\ldots,$		$T_{n-1}\}$
DCT-IV	$\{U_0,$	$U_1-U_0,$	$\ldots,$		$U_{n-1}-U_{n-2}\}$
DST-I	$\{U_0,$	$U_1,$	$\ldots,$		$U_{n-1}\}$
DST-II	$\{U_0,$	$U_1+U_0,$	$\ldots,$		$U_{n-1}+U_{n-2}\}$
DST-III	$\{U_0,$	$U_1,$	$\ldots,$	$U_{n-2},$	$\frac{1}{\sqrt{2}}U_{n-1}\}$
DST-IV	$\{U_0,$	$U_1+U_0,$	$\ldots,$		$U_{n-1}+U_{n-2}\}$

For each of the eight systems $\{Q_k(x)\}_{k=0}^n$ the above Table 2 lists the first n polynomials. To specify V_Q we have to also define the nodes $\{x_k\}_{k=1}^n$, or, equivalently, the last polynomial $Q_n(x)$, which is done in the second part of Table 2.

Table 2. Continuation. The last polynomial $Q_n(x)$.

	Q_n	zeros of Q_n
DCT-I	$xT_{n-1}-T_{n-2}$	$\{\cos(\frac{k\pi}{N-1})\}_0^{N-1}$
DCT-II	$U_n-2U_{n-1}+U_{n-2}$	$\{\cos(\frac{k\pi}{N})\}_0^{N-1}$
DCT-III	T_n	$\{\cos(\frac{(2k+1)\pi}{2N})\}_)^{N-1}$
DCT-IV	$2T_n$	$\{\cos(\frac{(2k+1)\pi}{2N})\}_)^{N-1}$
DST-I	U_n	$\{\cos(\frac{k\pi}{N+1})\}_1^N$
DST-II	$U_n+2U_{n-1}+U_{n-2}$	$\{\cos(\frac{k\pi}{N})\}_1^N$
DST-III	T_n	$\{\cos(\frac{(2k-1)\pi}{2N})\}_1^N$
DST-IV	$2T_n$	$\cos(\frac{(2k-1)\pi}{2N})\}_1^N$

Table 3.

DCT-I	$C_N^I=W_Q\cdot V_Q$	with	$W_Q=\sqrt{\frac{2}{N-1}}\mathrm{diag}(\frac{1}{\sqrt{2}},1,\ldots,1,\frac{1}{\sqrt{2}})$
DCT-II	$C_N^{II}=W_Q\cdot V_Q$	with	$W_Q=\sqrt{\frac{2}{N}}\mathrm{diag}(\frac{1}{\sqrt{2}},\cos(\frac{\pi}{2N}),\ldots,\cos(\frac{(N-1)\pi}{2N}))$
DCT-III	$C_N^{III}=W_Q\cdot V_Q$	with	$W_Q=\sqrt{\frac{2}{N}}\cdot I$
DCT-IV	$C_N^{IV}=W_Q\cdot V_Q$	with	$W_Q=\sqrt{\frac{2}{N}}\mathrm{diag}(\cos(\frac{\pi}{4N}),\cos(3\frac{\pi}{4N})\ldots,\cos(\frac{(2N-1)\pi}{4N}))$
DST-I	$S_N^I=W_Q\cdot V_Q$	with	$W_Q=\sqrt{\frac{2}{N+1}}\mathrm{diag}(\sin(\frac{\pi}{N+1}),\ldots,\sin(\frac{N\pi}{N+1}))$
DST-II	$S_N^{II}=W_Q\cdot V_Q$	with	$W_Q=\sqrt{\frac{2}{N}}\mathrm{diag}(\sin(\frac{\pi}{2N}),\ldots,\sin(\frac{(N-1)\pi}{2N}),\frac{1}{\sqrt{2}}\sin(\frac{\pi}{2}))$
DST-III	$S_N^{III}=W_Q\cdot V_Q$	with	$W_Q=\sqrt{\frac{2}{N}}\mathrm{diag}(\sin(\frac{\pi}{2N}),\sin(\frac{3\pi}{2N}),\ldots,\frac{1}{\sqrt{2}}\sin(\frac{(2N-1)\pi}{2N}))$
DST-IV	$S_N^{IV}=W_Q\cdot V_Q$	with	$W_Q=\sqrt{\frac{2}{N}}\mathrm{diag}(\sin(\frac{\pi}{4N}),\sin(\frac{3\pi}{4N}),\ldots,\sin(\frac{(2N-1)\pi}{4N}))$

Finally, we specify the corresponding confederate matrices H_Q.

Table 4.

DCT-I	$H_Q = \text{tridiag}\begin{bmatrix} & \frac{1}{\sqrt{2}} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{\sqrt{2}} & \\ 0 & & 0 & & 0 & \cdots & 0 & & 0 & & 0 \\ & \frac{1}{\sqrt{2}} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{\sqrt{2}} & \end{bmatrix}$
DCT-II	$H_Q = \text{tridiag}\begin{bmatrix} & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \\ \frac{1}{2} & & 0 & & 0 & \cdots & 0 & & 0 & & \frac{1}{2} \\ & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \end{bmatrix}$
DCT-III	$H_Q = \text{tridiag}\begin{bmatrix} & \frac{1}{\sqrt{2}} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \\ 0 & & 0 & & 0 & \cdots & 0 & & 0 & & 0 \\ & \frac{1}{\sqrt{2}} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \end{bmatrix}$
DCT-IV	$H_Q = \text{tridiag}\begin{bmatrix} & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \\ \frac{1}{2} & & 0 & & 0 & \cdots & 0 & & 0 & & -\frac{1}{2} \\ & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \end{bmatrix}$

Table 4. Continuation.

DST-I	$H_Q = \text{tridiag}\begin{bmatrix} & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \\ 0 & & 0 & & 0 & \cdots & 0 & & 0 & & 0 \\ & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \end{bmatrix}$
DST-II	$H_Q = \text{tridiag}\begin{bmatrix} & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \\ -\frac{1}{2} & & 0 & & 0 & \cdots & 0 & & 0 & & -\frac{1}{2} \\ & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \end{bmatrix}$
DST-III	$H_Q = \text{tridiag}\begin{bmatrix} & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \sqrt{\frac{1}{2}} & \\ 0 & & 0 & & 0 & \cdots & 0 & & 0 & & 0 \\ & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \sqrt{\frac{1}{2}} & \end{bmatrix}$
DST-IV	$H_Q = \text{tridiag}\begin{bmatrix} & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \\ -\frac{1}{2} & & 0 & & 0 & \cdots & 0 & & 0 & & \frac{1}{2} \\ & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} & & \frac{1}{2} & \end{bmatrix}$

All the proofs are straightforward and based on the well-known recurrence relations

$$T_0(x) = 1, \qquad T_1 = xT_0(x), \qquad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x).$$

$$U_0(x) = 1, \qquad U_1 = 2xU_0(x), \qquad U_k(x) = 2xU_{k-1}(x) - U_{k-2}(x).$$

In the second part of the paper we shall use the 8 displacement operators of the form

$$\nabla_{H_Q}(R) = H_Q R - R H_Q$$

with 8 Jacobi matrices listed in Table 4 to design real-arithmetic discrete cosine/sine transform based preconditioners. The expressions for the preconditioners will be obtained by using certain auxiliary formulas which it is convenient to present in the next section.

6 G.Strang-type preconditioners

Now we apply the technique developed in the first part of the paper to design a family of preconditioners for Toeplitz matrices. Consider a generating function of the form

$$a(x) = \sum_{k=-\infty}^{\infty} a_k z^k, \qquad a_k = a_{-k} \in \mathbf{R},$$

which we assume (a) to be from the Wiener class, i.e.,

$$\sum_{k=-\infty}^{\infty} |a_k| \leq \infty,$$

and (b) to have positive values on the unit circle,

$$f(z) > 0 \qquad |z| = 1.$$

As is well-known, these two conditions guarantee that all leading submatrices A_m $(m = 1, 2, \ldots)$ of the associated infinite real symmetric Toeplitz matrix

$$A = \left[\; a_{|i-j|} \; \right] = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots \\ a_1 & a_0 & a_1 & a_2 & \ddots \\ a_2 & a_1 & a_0 & a_1 & \ddots \\ a_3 & a_2 & a_1 & a_0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

are positive definite : $A_m > 0$. Our next goal is to construct for A_m (assuming that m is sufficiently large) a good preconditioner from the class $\mathcal{K}_Q =$ Ker ∇_{H_Q}, where H_Q is one of the 8 matrices in Table 4. The first property we would like such a preconditioner $S_Q(A_m)$ to satisfy is the following, satisfied by the classical circulant G.Strang preconditioner.

Property 4. *For any $\varepsilon > 0$ there exist $M > 0$ so that for $m > M$ the spectrum of $S_Q(A_m)$ lies in the interval* $[\min_{|z|=1} a(z) - \varepsilon, \max_{|z|=1} a(z) + \varepsilon]$.

Thus to obtain explicit formulas for $S_Q(A_m)$ we need to to apply Corollary 4.3 for each of the 8 cases, and to obtain the description of the spectrum of $S_Q(A_m)$. This is done in the next statement.

Corollary 6.1 *Let H_Q be one of the matrices in Table 4, $\mathcal{K}_Q =$ Ker ∇_{H_Q}, and let*

$$R = \sum_{k=0}^{n-1} r_k Q_k(H_Q) \tag{6.1}$$

be a decomposition of $R \in \mathcal{K}_Q$ with respect to a basis

$$\{Q_k(H_Q), k = 0, 1, ..., n-1\} \tag{6.2}$$

in $\mathcal{K}_Q$. Then

$$T_Q R T_Q^T = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \tag{6.3}$$

where T_Q is the corresponding discrete transform from Table 1, and

$$\begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix} = W_Q^{-1} T_Q \begin{bmatrix} r_0 \\ \vdots \\ r_{n-1} \end{bmatrix}, \tag{6.4}$$

where W_Q is the corresponding weight matrix from Table 3.

Formula (6.4) and explicit expressions for V_Q and W_Q listed in Tables 2 and 3, show that if we will define

$$S_Q(A) = \sum_{j=0}^{m-1} r_j \cdot Q_j(H_Q), \tag{6.5}$$

as in Table 5, then the eigenvalues of $S_Q(A_m)$ will have the form shown in Table 6.

Table 5. Definition of the G.Strang-type preconditioners. The coefficients of the decomposition (6.5) of $S_Q(A)$.

	r_0	r_1	r_2	$\cdots$	r_{m-3}	r_{m-2}	r_{m-1}
DCT-I	$\sqrt{2}\cdot a_0$	$2a_1$	$2a_2$	$\cdots$	$2a_{m-3}$	$2a_{m-2}$	$2\sqrt{2}\cdot a_{m-1}$
DCT-II	a_0+a_1	a_1+a_2	a_2+a_3	$\cdots$	$a_{m-3}+a_{m-2}$	$a_{m-2}+a_{m-1}$	a_{m-1}
DCT-III	$\sqrt{2}\cdot a_0$	$2a_1$	$2a_2$	$\cdots$	$2a_{m-3}$	$2a_{m-2}$	$2a_{m-1}$
DCT-IV	a_0+a_1	a_1+a_2	a_2+a_3	$\cdots$	$a_{m-3}+a_{m-2}$	$a_{m-2}+a_{m-1}$	a_{m-1}
DST-I	a_0-a_2	a_1-a_3	a_2-a_4	$\cdots$	$a_{m-3}-a_{m-1}$	a_{m-2}	a_{m-1}
DST-II	a_0-a_1	a_1-a_2	a_2-a_3	$\cdots$	$a_{m-3}-a_{m-2}$	$a_{m-2}-a_{m-1}$	a_{m-1}
DST-III	a_0-a_2	a_1-a_3	a_2-a_4	$\cdots$	$a_{m-3}-a_{m-1}$	a_{m-2}	$\sqrt{2}a_{m-1}$
DST-IV	a_0-a_1	a_1-a_2	a_2-a_3	$\cdots$	$a_{m-3}-a_{m-2}$	$a_{m-2}-a_{m-1}$	a_{m-1}

Table 6. Eigenvalues $\{\lambda_1, \ldots, \lambda_m\}$ of $S_Q(A)$.

DCT-I	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(k-1)j\pi}{m-1}$	$=$	$a_m(z_k)$	where	$z_k = e^{\frac{k-1}{m-1}\pi i}$
DCT-II	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(k-1)j\pi}{m}$	$=$	$a_m(z_k)$	where	$z_k = e^{\frac{k-1}{m}\pi i}$
DCT-III	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(2k-1)j\pi}{2m}$	$=$	$a_m(z_k)$	where	$z_k = e^{\frac{2k-1}{2m}\pi i}$
DCT-IV	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(2k-1)j\pi}{2m}$	$=$	$a_m(z_k)$	where	$z_k = e^{\frac{2k-1}{2m}\pi i}$
DST-I	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{kj\pi}{m+1}$	$=$	$a_m(z_k)$	where	$z_k = e^{\frac{k}{m+1}\pi i}$
DST-II	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{kj\pi}{m}$	$=$	$a_m(z_k)$	where	$z_k = e^{\frac{k}{m}\pi i}$
DST-III	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(2k-1)j\pi}{2m}$	$=$	$a_m(z_k)$	where	$z_k = e^{\frac{2k-1}{2m}\pi i}$
DST-IV	$\lambda_k = a_0 + 2\sum_{j=1}^{m-1} a_j \cdot \cos\frac{(2k-1)j\pi}{2m}$	$=$	$a_m(z_k)$	where	$z_k = e^{\frac{2k-1}{2m}\pi i}$

(In Table 6 we list expressions for all 8 cases, because we shall use them in our arguments below.) Thus, the eigenvalues $\{\lambda_k\}$ of $S_Q(A)$ are the values of a truncated function

$$a_m(z) = \sum_{k=-m+1}^{m+1} a_k z^k$$

at certain points on the unit circle, specified in Table 6. Since $a(x)$ is in the Wiener class, $a_m(x)$ is its approximation, so the Property 4 (defined in this section above) holds.

Moreover the following two properties are easily deduced from the property 4.

Property 5. *All G.Strang-type preconditioners $S_Q(A)$ specified in Table 5 are positive definite matrices for sufficiently large m.*

Property 6. *For all G.Strang-type preconditioners in Table 5 we have that $\|S_Q(A)\|_2$ and $\|S_Q(A)^{-1}\|_2$ are uniformly bounded independently of m.*

Note that the DST-I-based preconditioner was designed earlier in [BB90] and it was also discussed in [BK95] [H95].

Summarizing, in this section we presented explicit formulas for G.Strang-type preconditioners, and proved for them the properties 4 - 6. Moreover, the properties 1 - 2 stated in the Introduction are also trivially satisfied. It remains only to establish the property 3, crucial for the rapid convergence of the PCGM. This property will be proved in Sec. 8 below, using another description of the new preconditioners given next.

7 The new preconditioners are Toeplitz-plus-Hankel-like matrices

7.1. The classical G.Strang (Toeplitz-plus-Toeplitz) preconditioner. We called new preconditioners as G.Strang-like preconditioners, a justification for this nomenclature is offered next. In [S86] G.Strang proposed a circulant preconditioner,

$$S(A) = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_2 & a_1 \\ a_1 & a_0 & a_1 & \ddots & & a_2 \\ a_2 & a_1 & a_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_1 & a_2 \\ a_2 & & \ddots & a_1 & a_0 & a_1 \\ a_1 & a_2 & \cdots & a_2 & a_1 & a_0 \end{bmatrix}$$

obtained by copying first $[m/2]$ diagonals of $A = \begin{bmatrix} a_{|i-j|} \end{bmatrix}$. In fact, this preconditioner can be seen as a Toeplitz-plus-Toeplitz matrix,

$$S(A) = A + T,$$

where A is the given Toeplitz matrix, and the first column of a second Toeplitz term T is given by $\begin{bmatrix} 0 & \cdots & 0 & \cdots & a_2 - a_{m-2} & a_1 - a_{m-1} \end{bmatrix}^T$. In fact, many favorable properties of $S(A)$ can be explained by the fact that the entries of the central diagonals of A now occupy corner positions in T. In case when the generating function $a(z) = \sum_{k=-\infty}^{\infty} a_k \cdot z^k$ is from the Wiener class, only first few coefficients are large, implying that $T = A_{lr} + A_{sn}$ is a sum of a low rank and a small norm matrices, a property implying the usefulness of

$$S(A) = A + A_{lr} + A_{sn} \tag{7.1}$$

as a preconditioner for A. It turns out that all 8 G.Strang-type preconditioners $S_Q(A)$, considered above are Toeplitz-plus-Hankel-like matrices (formally defined below), a fact allowing us to use the above low-rank-small-norm-perturbation argument to prove the favorable properties of $S_Q(A)$ as preconditioners for A.

7.2. Toeplitz-plus-Hankel-like matrices. Recall that matrices R with a low ∇_{H_Q}-displacement rank have been called *polynomial Hankel* matrices. Since all 8 matrices H_Q in Table 4 correspond to the Chebyshev-like polynomial systems listed in Table 2, we could refer to such R as *Chebyshev-Hankel* matrices. It can be checked, however, that if H_Q is defined as, for example, in the line DST-I of Table 4, then for any sum $T+H$ of a Toeplitz $T = \begin{bmatrix} t_{i-j} \end{bmatrix}$ and a Hankel $H = \begin{bmatrix} h_{i+j-2} \end{bmatrix}$ matrices we have

$$\nabla_{H_Q}(T + H) = \tag{7.2}$$

$$\frac{1}{2}\cdot\left(\left[\begin{array}{c|ccc|c} 0 & -t_2 & \cdots & -t_{m-1} & 0 \\ \hline t_2 & & & & t_{m-1} \\ \vdots & & 0 & & \vdots \\ t_{m-1} & & & & t_2 \\ \hline 0 & -t_{m-1} & \cdots & -t_2 & 0 \end{array}\right] + \left[\begin{array}{c|ccc|c} 0 & -h_0 & \cdots & -h_{m-3} & h_{2m-2}-h_m \\ \hline h_0 & & & & h_{m+1} \\ \vdots & & 0 & & \vdots \\ h_{m-3} & & & & h_{2m-2} \\ \hline h_{m-2}-h_m & -h_{m+1} & \cdots & -h_{2m-2} & 0 \end{array}\right]\right).$$

In our terminology, the ∇_{H_Q}-displacement rank of $T+H$ does not exceed 4. This fact was observed and used in [HJR88] and [GK89] to develop fast algorithms for inversion of Toeplitz-plus-Hankel matrices. In [GKO95] we introduced (and suggested, for the first time, fast algorithms for) the more general class of Toeplitz-plus-Hankel-like matrices, defined as having low (not just $\alpha \leq 4$) ∇_{H_Q}-displacement rank. Clearly (cf. with [GKO95]), the other choices for H_Q in table 4 can be used to define the *same* class of Toeplitz-plus-Hankel-like matrices (the actual displacement rank may vary, depending upon a particular H_Q, but it remains low). Summarizing, there are two nomenclatures (i.e. Chebyshev Hankel-like and Toeplitz-plus-Hankel-like matrices) for the same class of structured matrices.

7.3. Toeplitz-plus-Hankel-like representations for $S_Q(A)$. Since all the preconditioners $S_Q(A)$ in Table 5 belong to the kernel of the corresponding $\nabla_{H_Q}(\cdot)$, they clearly belong to the above class of Toeplitz-plus-Hankel-like matrices. In fact, each of them can even be represented as

$$S_Q(A) = A + H + B, \tag{7.3}$$

where A is the given Toeplitz matrix, H is a certain Hankel matrix, and B is a certain "border" matrix, having nonzero entries only in its first and last rows and columns[2].

Table 7. A Hankel part and a "border" part of $S_Q(A)$.

	H	B
DCT-I	$\begin{bmatrix} a_0 & a_1 & \cdots & a_{m-2} & 2a_{m-1} \\ a_1 & a_2 & \iddots & \iddots & a_{m-2} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ a_{m-2} & \iddots & \iddots & a_2 & a_1 \\ 2a_{m-1} & a_{m-2} & \cdots & a_1 & a_0 \end{bmatrix}$	$(\sqrt{2}-2)\cdot\left[\begin{array}{c\|ccc\|c} -\frac{a_0}{\sqrt{2}-2} & a_1 & \cdots & a_{m-2} & -\frac{a_{m-1}}{\sqrt{2}-2} \\ \hline a_1 & & & & a_{m-1} \\ \vdots & & 0 & & \vdots \\ a_{m-2} & & & & a_1 \\ \hline -\frac{a_{m-1}}{\sqrt{2}-2} & a_{m-2} & \cdots & a_1 & -\frac{a_0}{\sqrt{2}-2} \end{array}\right]$
DCT-II	$\begin{bmatrix} a_1 & a_2 & \cdots & a_{m-1} & 0 \\ a_2 & a_3 & \iddots & \iddots & a_{m-1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ a_{m-1} & \iddots & \iddots & a_3 & a_2 \\ 0 & a_{m-1} & \cdots & a_2 & a_1 \end{bmatrix}$	0
DCT-III	$\begin{bmatrix} a_0 & a_1 & \cdots & a_{m-2} & a_{m-1} \\ a_1 & a_2 & \iddots & \iddots & 0 \\ \vdots & \iddots & \iddots & \iddots & -a_{m-1} \\ a_{m-2} & \iddots & \iddots & \iddots & \vdots \\ a_{m-1} & 0 & -a_{m-1} & \cdots & -a_2 \end{bmatrix}$	$(\sqrt{2}-2)\cdot\left[\begin{array}{c\|cccc} -\frac{a_0}{\sqrt{2}-2} & a_1 & \cdots & a_{m-2} & a_{m-1} \\ \hline a_1 & & & & \\ \vdots & & & 0 & \\ a_{m-2} & & & & \\ a_{m-1} & & & & \end{array}\right]$
DCT-IV	$\begin{bmatrix} a_1 & a_2 & \cdots & a_{m-1} & 0 \\ a_2 & a_3 & \iddots & \iddots & -a_{m-1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ a_{m-1} & \iddots & \iddots & -a_3 & -a_2 \\ 0 & -a_{m-1} & \cdots & -a_2 & -a_1 \end{bmatrix}$	0

Table 7. A Hankel part and a "border" part of $S_Q(A)$. Continuation.

[2]A reader should be warned that such a specific representation is not valid for arbitrary Toeplitz-plus-Hankel-like matrices.

<table>
<tr><th></th><th>H</th><th>B</th></tr>
<tr><td>DST-I</td><td>$\begin{bmatrix} -a_2 & \cdots & -a_{m-1} & 0 & 0 \\ \vdots & \iddots & \iddots & \iddots & 0 \\ -a_{m-1} & \iddots & \iddots & \iddots & -a_{m-1} \\ 0 & \iddots & \iddots & \iddots & \vdots \\ 0 & 0 & -a_{m-1} & \cdots & -a_2 \end{bmatrix}$</td><td>$0$</td></tr>
<tr><td>DST-II</td><td>$\begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{m-1} & 0 \\ -a_2 & -a_3 & \iddots & \iddots & -a_{m-1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ -a_{m-1} & \iddots & \iddots & -a_3 & -a_2 \\ 0 & -a_{m-1} & \cdots & -a_2 & -a_1 \end{bmatrix}$</td><td>$0$</td></tr>
<tr><td>DST-III</td><td>$\begin{bmatrix} -a_2 & \cdots & -a_{m-1} & 0 & a_{m-1} \\ \vdots & & \iddots & \iddots & a_{m-2} \\ -a_{m-1} & \iddots & \iddots & \iddots & \vdots \\ 0 & \iddots & \iddots & a_2 & a_1 \\ a_{m-1} & a_{m-2} & \cdots & a_1 & a_0 \end{bmatrix}$</td><td>$\frac{1}{1+\sqrt{2}} \cdot \left[\begin{array}{cccc|c} & & 0 & & -a_{m-1} \\ & & & & -a_{m-2} \\ & & & & \vdots \\ & & & & -a_1 \\ \hline -a_{m-1} & -a_{m-2} & \cdots & -a_1 & -(1+\frac{1}{\sqrt{2}})a_0 \end{array} \right]$</td></tr>
<tr><td>DST-IV</td><td>$\begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{m-1} & 0 \\ -a_2 & -a_3 & \iddots & \iddots & a_{m-1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ -a_{m-1} & \iddots & \iddots & a_3 & a_2 \\ 0 & a_{m-1} & \cdots & a_2 & a_1 \end{bmatrix}$</td><td>$0$</td></tr>
</table>

The proof of the fact that $S_Q(A)$ has the form (7.3) where A is the given Toeplitz matrix, and H, B are specified in Table 7, is based on the following observations.

- The fact that $A + H + B \in \mathcal{K}_Q =$ Ker ∇_{H_Q} can be easily checked by inspection.
- C.Di Fiore and P.Zellini showed in [DFZ95] that

$$Q_k(H_Q) \cdot e_1 = Q_0 \cdot e_{k+1}, \tag{7.4}$$

  and used this fact to derive inversion formulas for structured matrices. As was mentioned in Sec. 4, 6, matrices $\{Q_k(H_Q)\}$ form a basis in Ker ∇_{H_Q} (because of the nice property (7.4) we suggest to call call it Di Fiore-Zellini basis), and since the first column of the matrix $A + H + B$ coincides with the first column of $S_Q(A)$ given in Table 5, the representations (7.3) follow.

Recall that we started this section saying that all eight $S_Q(A)$ are the (discrete-trigonometric-transform) analogs of the Strang preconditioner S. Indeed, the results in Table 7 show that the form (7.3) for $S_Q(A)$ is similar to that in (7.1) for $S(A)$. Indeed, both the Strang circulant preconditioner $S(A)$ and all of $S_Q(A)$ are constructed by adding to the given Toeplitz matrix A a matrix, in which the entries of the central diagonals of A now occupy the corner locations. This fact is used next to prove the useful properties of $S_Q(A)$.

8 Clustering of the spectrum of $S_Q(A)^{-1}A$

Here we establish the crucial property 3 for all 8 G.Strang-like preconditioners $S_Q(A)$ under the standard Wiener class assumption. For the preconditioner $S_Q(A)$ corresponding to the DST-I, this property was established in [BB90] and below we adapt their arguments for the other seven $S_Q(A)$. Since $S_Q(A)^{-1}A = I - S_Q(A)^{-1}(H + B)$ with H, B specified in table 6, it is sufficient to show that the spectrum of $S_Q(A)^{-1}(H+B)$ is clustered around 0. Let $\varepsilon > 0$ be fixed, choose N such that $\sum_{k=N+1}^{\infty} |a_k| < \varepsilon$. Then we can asplit

$$H + B = A_{lr} + A_{sn}, \tag{8.1}$$

by taking out in A_{lr} antidiagonals of $H+B$ with the entries $a_0, a_1, \ldots, a_N$. Then the 2-norm of the second matrix in (8.1) can be bounded by 2ε (cf. with [BB90]). Hence by the Cauchy interlace theorem, the eigenvalues of $(H+B)$ are clustered around zero, except at most $s =$ rank A_{lr} outliers. Applying the Courant-Fischer theorem to the matrix $S_Q(A)^{-1}(H+B)$, we obtain

$$\lambda_k\{(S_Q(A)^{-1}(H+B)\} < \frac{\lambda_k\{H+B\}}{\min_{|z|=1} f(z)}$$

implying that the Property 3 holds.

9 T.Chan-type preconditioners

Here we specify another family of preconditioners $C_Q(A)$, defined by

$$\|C_Q(A) - A\|_F = \min_{R \in \mathcal{K}_Q} \|R - A\|_F,$$

i.e., the optimal Frobenius-norm approximants of A in $\mathcal{K}_Q =$ Ker ∇_{H_Q} (for the circulant case, $H_Q = Z_1$, such preconditioner was proposed by T.Chan [C88]).

Recall that we designed all G.Strang-type preconditioners $S_Q(A)$ using a representation of the form (6.5). It turns out that the same basis is convenient for writing down the formulas also for the T.Chan-type preconditioners $C_Q(A)$, and especially for the analysis of the clustering property in Sec. 9. In order to obtain the coefficients in

$$C_Q(A) = \sum_{k=1}^{m} r_k Q_{k-1}(H_Q), \tag{9.1}$$

we solve a linear system of equations

$$\frac{\partial}{\partial r_k} \|C_Q(A) - A\|_F = 0 \qquad (k = 1, 2, \ldots, m). \tag{9.2}$$

To solve (9.2) we found the entries of matrices $\{Q_k(H_Q)\}$ in all eight cases as follows. The entries of the first column of each $Q_k(H_Q)$ are given by (7.4). The entries of the other columns can be recursively computed using the fact that $Q_k(H_Q) \in \mathcal{K}_Q =$ Ker ∇_{H_Q}. For example, for $n = 6$ we have : $Q_0(H_Q) = Q_0 I$ (with $Q_0 = \frac{1}{\sqrt{2}}$ for DCT-I and DCT-III, and $Q_0 = 1$ in the other six cases), $Q_1(H_Q) = H_Q$ (see, e.g., Table 4), and

Table 8. First-column bases.

	$Q_2(H_Q)$	$Q_3(H_Q)$	$Q_4(H_Q)$
DCT-I	$\begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \end{bmatrix}$
DCT-II	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 \\ 1 & -1 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & 0 \\ 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
DCT-III	$\begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} & 0 & 0 \end{bmatrix}$
DCT-IV	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 \\ 1 & -1 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -2 \\ 0 & 0 & 1 & -2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -2 \\ 1 & -1 & 1 & -2 & 2 \\ 0 & 1 & -2 & 2 & -2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 1 & -2 & 2 \\ 0 & 1 & -2 & 2 & -2 \\ 1 & -2 & 2 & -2 & 2 \end{bmatrix}$

Table 8. First-column bases. Continuation.

	$Q_2(H_Q)$	$Q_3(H_Q)$	$Q_4(H_Q)$
DST-I	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
DST-II	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
DST-III	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & \sqrt{2} \\ 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & \sqrt{2} \\ 0 & 1 & 0 & 2 & 0 \\ 1 & 0 & 2 & 0 & \sqrt{2} \\ 0 & \sqrt{2} & 0 & \sqrt{2} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & \sqrt{2} & 0 & 1 \\ 0 & \sqrt{2} & 0 & \sqrt{2} & 0 \\ 1 & 0 & 1 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$
DST-IV	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 1 & 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 2 & 2 \\ 0 & 1 & 2 & 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 2 & 2 \\ 1 & 2 & 2 & 2 & 2 \end{bmatrix}$

Using a particular form of $Q_k(H_Q)$ shown in Table 7, we obtain from (9.2) that the desired coefficients in (9.1) are then obtained from the given Toeplitz matrix $A = \left[\ a_{|i-j|}\ \right]$ by

$$\begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} = G_Q \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}, \tag{9.3}$$

where the matrix G_Q has the simple structure shown in the next Table.

Table 9. Definition of the T.Chan-type preconditioner $C_Q(A_m)$. Matrix G_Q for (9.3).

DCT-I	$G = \frac{1}{(m-1)^2} \cdot$ diag $\{\sqrt{2}, 2, 2, \ldots 2, \sqrt{2}\}(D+E+L+U)$, with the terms specified by (9.4), (9.5), (9.6), (9.7)
DCT-II	$\frac{1}{m^2} \begin{bmatrix} m^2 & (m-1)(m-2) & -2(m-2) & \cdots & -4 & -2 \\ 0 & m^2-(m-2) & (m-2)(m-2) & \ddots & -4 & -2 \\ 0 & 2 & m^2-2(m-2) & \ddots & -4 & -2 \\ 0 & 2 & 4 & \ddots & 2(m-2) & -2 \\ 0 & 2 & 4 & \ddots & m^2-(m-2)(m-2) & (m-2) \\ 0 & 2 & 4 & \cdots & 2(m-2) & m^2-(m-1)(m-2) \end{bmatrix}$
DCT-III	$\frac{1}{m} \begin{bmatrix} \sqrt{2}m & & & & & \\ & 2(m+\sqrt{2}-2) & & & \mathbf{0} & \\ & & 2(m+\sqrt{2}-3) & & & \\ & & & \ddots & & \\ & \mathbf{0} & & & 2(\sqrt{2}+1) & \\ & & & & & 2\sqrt{2} \end{bmatrix}$
DCT-IV	$\frac{1}{m} \begin{bmatrix} m & m-1 & & & & \\ & m-1 & m-2 & & \mathbf{0} & \\ & & m-2 & \ddots & & \\ & & & \ddots & 2 & \\ & \mathbf{0} & & & 2 & 1 \\ & & & & & 1 \end{bmatrix}$

$$D = \text{ diag } \left((m-1)^2, \boxed{2\sqrt{2}(m-1)+(m-3)(m-3)}, \boxed{2\sqrt{2}(m-1)+(m-3)(m-4)}, \ldots \right. \tag{9.4}$$

$$\left. \ldots, \boxed{2\sqrt{2}(m-1)+2(m-3)}, \boxed{2\sqrt{2}(m-1)+(m-3)}, \boxed{2\sqrt{2}(m-1)}, (2m-3)\right),$$

(a recursion for the $2, 3, \ldots, m-2, m-1$ entries is apparent.)

$$E = -2\sqrt{2} \text{ toeplitz } (\left[\ 1,\ \ 0,\ \ 1,\ \ 0,\ \ 1,\ \ 0,\ \ \ldots\ \right]) \cdot \text{ diag } (\left[\ 0,\ \ 1,\ \ 1,\ \ \ldots, 1,\ \ 0\ \right]) \tag{9.5}$$

Here we follow the MATLAB notations, where toeplitz (c, r) denotes the Toeplitz matrix with the first column c and the first row r. toeplitz (c) denotes the symmetric Toeplitz matrix with the first column c.

$$L = \text{ toeplitz } (\left[\ 0,\ \ 0,\ \ 1,\ \ 0,\ \ 1,\ \ 0,\ \ 1,\ \ 0,\ \ \ldots\ \right], \left[\ 0,\ \ 0,\ \ 0,\ \ \ldots\ \right]) \times \tag{9.6}$$

$$
\begin{aligned}
&\operatorname{diag}\left(\begin{bmatrix} 0, & 2\cdot 2, & 2\cdot 3, & \ldots, & 2\cdot(m-2), & 0, & 0 \end{bmatrix}\right) \\
U = {}&\operatorname{toeplitz}\left(\begin{bmatrix} 0, & 0, & 0 & \ldots \end{bmatrix}, \begin{bmatrix} 0, & 0, & 1, & 0, & 1, & 0, & 1, & 0, & \ldots \end{bmatrix}\right)\times \\
&\operatorname{diag}\left(\begin{bmatrix} 0, & 0, & -2(m-4), & -2(m-3), & \ldots, & -4, & -2, & 0, & -1 \end{bmatrix}\right).
\end{aligned}
\tag{9.7}
$$

Table 9. Continuation. Definition of the T.Chan-type preconditioner $C_Q(A_m)$. Matrix G_Q for (9.3).

DST-I	$\frac{1}{m+1}\begin{bmatrix} m+1 & 0 & -(m-2) & & & & \\ & m+1 & 0 & -(m-3) & & & \\ & & m & \ddots & \ddots & & \\ & & & \ddots & 0 & -2 & \\ & \mathbf{0} & & & 5 & 0 & -1 \\ & & & & & 4 & 0 \\ & & & & & & 3 \end{bmatrix}$
DST-II	$\frac{1}{m^2}\begin{bmatrix} m^2 & -(m-1)(m-2) & -2(m-2) & \cdots & (-1)^{m-2}4 & (-1)^{m-1} \\ 0 & m^2-(m-2) & -(m-2)(m-2) & \ddots & \vdots & \vdots \\ 0 & -2 & m^2-2(m-2) & \ddots & -4 & 2 \\ 0 & 2 & -4 & \ddots & -2(m-2) & -2 \\ 0 & \vdots & \vdots & \ddots & m^2-(m-2)(m-2) & -(m-2) \\ 0 & (-1)^m 2 & (-1)^{m-1}4 & \cdots & -2(m-2) & m^2-(m-1)(m-2) \end{bmatrix}$
DST-III	$\frac{1}{m}\begin{bmatrix} m & 0 & -(-3+\sqrt{2}) & & & & \\ & m-2+\sqrt{2} & 0 & \ddots & & & \\ & & m-3+\sqrt{2} & \ddots & -(2+\sqrt{2}) & & \\ & & & \ddots & 0 & -(1+\sqrt{2}) & \\ & \mathbf{0} & & & 2+\sqrt{2} & 0 & -\sqrt{2} \\ & & & & & 1+\sqrt{2} & 0 \\ & & & & & & 2 \end{bmatrix}$
DST-IV	$\frac{1}{m}\begin{bmatrix} m & -(m-1) & & & & \\ & m-1 & -(m-2) & & \mathbf{0} & \\ & & m-2 & \ddots & & \\ & & & \ddots & -2 & \\ & \mathbf{0} & & & 2 & -1 \\ & & & & & 1 \end{bmatrix}$

Formula (9.3) along with the data in Table10 show that all T.Chan-type preconditioners trivially satisfy the properties 1, 2. In the next section we establish for them the property 3.

10 Clustering of eigenvalues of $C_Q(A)^{-1}A$.

10.1. Clustering. Here we establish for all 8 preconditioners $C_Q(A)$ the property 3 under the Wiener class assumption, by showing that the spectra of $S_Q(A)$ and $C_Q(A)$ are assimptotically the same (for the classical G.Strang and T.Chan circulant preconditioners this propery was established in [C89]).

Proposition 10.1 *Let A_m be a finite section of a single-infinite Toeplitz matrix, $\{Q_k\}_{k=0}^{m-1}$ be one of the eight polynomial systems in Table 2, and let $S_Q(A), C_Q(A) \in \mathbf{R}^{n\times n}$ denote the corresponding G.Strang-type and the T.Chan-type preconditioners, respectively. Then*

$$\lim_{m\to\infty} \|C_Q(A) - S_Q(A)\|_2 = 0,$$

where $\|\cdot\|_2$ denotes the spectral norm in $\mathbf{R}^{m\times m}$.

Proof. Since both $C_Q(A)$ and $S_Q(A)$ belong to $\mathcal{K}_Q$, they are both diagonalized by the corresponding discrete-trigonometric transform matrix T_Q, see, e.g., Proposition 6.1. Since T_Q is an orthogonal matrix (i.e, one of the 8 orthogonal matrices displayed in Table 1), we have essentially to establish the convergence to zero of the eigenvalues of $S_Q(A) - C_Q(A)$:

$$\lim_{m\to\infty} \lambda_k(S_Q(A) - C_Q(A)) = 0 \qquad (k = 1, 2, ..., m).$$

Again, by Proposition 6.1 the eigenvalues of $S_Q(A)$ and $C_Q(A)$ can be obtained from the coefficients $\{r_k\}$ in the representation (6.1) for these matrices by using (6.4). As shown in Sec. 9, for the $C_Q(A)$ these coefficients are given by

$$\begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} = G_Q \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}, \tag{10.1}$$

where the matrices G_Q are listed in Table10. For convenience we next rewrite the results of Table 5 in a similar manner. Moreover, the coefficients in the representation $S_Q(A) = \sum_{k=0}^{m-1} r_k C_Q(H_Q^T)$ are obtained by

$$\begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} = R_Q \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}, \tag{10.2}$$

where matrices R_Q are specified in Table 9.

Table 10. Definition of $S_Q(A_m)$. The matrix R_Q in (10.2).

DCT-I	$\begin{bmatrix} \sqrt{2} & & & & & \\ & 2 & & & & \\ & & 2 & & & \\ & & & \ddots & & \\ & & & & 2 & \\ & & & & & 2\sqrt{2} \end{bmatrix}$	DST-I	$\begin{bmatrix} 1 & 0 & -1 & & & & \\ & 1 & 0 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & -1 & \\ & & & & 1 & 0 & -1 \\ & & & & & 1 & 0 \\ & & & & & & 1 \end{bmatrix}$
DCT-II	$\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & 1 & \\ & & & & 1 & 1 \end{bmatrix}$	DST-II	$\begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & -1 & \\ & & & & 1 & -1 \end{bmatrix}$

DCT-III	$\begin{bmatrix} \sqrt{2} & & & & & \\ & 2 & & & & \\ & & 2 & & & \\ & & & \ddots & & \\ & & & & 2 & \\ & & & & & 2 \end{bmatrix}$	DST-III	$\begin{bmatrix} 1 & 0 & -1 & & & & \\ & 1 & 0 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & -1 & \\ & & & & 1 & 0 & -1 \\ & & & & & 1 & 0 \\ & & & & & & \sqrt{2} \end{bmatrix}$
DCT-IV	$\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & 1 & \\ & & & & 1 & 1 \end{bmatrix}$	DST-IV	$\begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & -1 & \\ & & & & 1 & -1 \end{bmatrix}$

By comparing (6.4), (10.1) and (10.2) we have :

$$\begin{bmatrix} \lambda_1(S_Q(A) - C_Q(A)) \\ \vdots \\ \lambda_m(S_Q(A) - C_Q(A)) \end{bmatrix} = V_Q \cdot (R_Q - G_Q) \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}, \tag{10.3}$$

where the matrices R_Q and G_Q are displayed in Tables 10 and 11, respectively. Recall that not all of the eight matrices V_Q have uniformly bounded entries, see, e.g., Table 2. Therefore it is more convenient to rewrite (10.3) as

$$\begin{bmatrix} \lambda_1(S_Q(A) - C_Q(A)) \\ \vdots \\ \lambda_m(S_Q(A) - C_Q(A)) \end{bmatrix} = (V_Q R_Q) \cdot (I - R_Q^{-1} G_Q) D^{-1} \cdot (D \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix}), \tag{10.4}$$

where $D =$ diag $(\frac{1}{m}, \frac{2}{m}, ..., \frac{m-1}{m}, 1)$. Now we can prove the statement of the proposition, i.e. that the entries on the left-hand side of (10.4) tend to zero by making three following observations for three factors on the right-hand side of (10.4).

1. **Left factor.** The entries of the matrix $V_Q R_Q$ are uniformly bounded independently of m.

2. **Middle factor.** The column sums of the matrix $(I - R_Q^{-1} G_Q) D^{-1}$ have uniformly bounded column sums.

3. **Right factor.** If $f(z) = \sum_{k=-\infty}^{\infty} a_k z^k$ is from the Wiener class, then $\forall \varepsilon > 0\ \exists N > 1$ such that $\forall M > N$ we have
$$\sum_{k=0}^{M} \frac{k}{M} |a_k| < \varepsilon.$$

The first observation can be deduced from the comparison of (10.2) and the Table 6, showing that $V_Q \cdot R_Q$ is a "cosine" matrix.

The assertion in the second observation is deduced from the particular form of matrices R_Q and G_Q displayed in Tables 9 and 12. The arguments are immediate in the cases DCT-III, DCT-IV, DST-III, DST-IV, and they are not much more involved in the case of DST-I. In the cases of DCT-I, DCT-II and DST-II one has to split G_Q into three parts : bidiagonal, upper and lower triangular, and for each of the corresponding parts of $(I - R_Q^{-1} G_Q) D^{-1}$ the statement is easily deduced. The third observation is immediate (cf. with [C89]).

Thus, Proposition 10.1 is now proved, and it implies the property 3 using standard arguments (cf. with [C89]).

10.2. Transformation-to-Cauchy-like and the Tyrtyshnikov property. In the previous section we proved the properties 1-3 for $C_Q(A_m)$. The properties 4 and 6 (formulated in Sec. 6) also follow from Prop. 10.1. Here we show that property 5 also holds, and, moreover we prove that for any of 8 T.Chan-type preconditioners the following *Tyrtyshnikov property* holds independently of m

$$\lambda_{min}(A_m) \leq \lambda_{min}(C_Q(A_m)) \leq \lambda_{max}(C_Q(A_m)) \leq \lambda_{max}(A_m). \tag{10.5}$$

For the circulant T.Chan preconditioner such a property was proved in [T92] and [CJY91].

To prove (10.5) we shall use our definitions in Sec. 2. Observe that since the Frobenius norm in $\mathbf{R}^{m\times m}$ generates the inner product (2.6), we have that the ∇_{H_Q}-generator of A_m is given by $\{G_Q, J, C_Q(A_m)\}$ in

$$H_Q A_m - A_m H_Q = G_Q J G_Q^T.$$

The particular form of $m \times 4$ matrix G_Q and 4×4 matrix J is not relevant at the moment (for each H_Q of Table 4 they can be easily written down as, for example, in (7.2). It is important that the corresponding T.Chan-type preconditioner describes the kernel component of A_m, i.e., the third matrix in its ∇_{H_Q}-generator. Furthermore, specifying the "transformation-to-Cauchy-like" prop. 4.1 to our settings here we obtain the results displayed in the following Table 11.

Table 11. Transformation-to-Cauchy-like.

	Toeplitz	$\to$	Cauchy-like
Matrix	A_m	$\to$	$T_Q \cdot A_m \cdot T_Q^T$
Generator	G_Q	$\to$	$T_Q \cdot G_Q$
	J	$\to$	J
	$C_Q(A_m)$	$\to$	$T_Q \cdot C_Q(A_m) \cdot T_Q^T$

The inequality (10.5) now follows from the following two observations. First, since T_Q is orthogonal, the spectra of A_m and $T_Q A_m T_Q^T$ and of $C_Q(A_m)$ and $T_Q C_Q(A_m) T_Q^T$ are, respectively, the same. Second, since the Frobenius norm is unitary-equivalent, we have that the diagonal matrix $T_Q \cdot C_Q(A_m) \cdot T_Q^T$ is the optimal Frobenius-norm diagonal approximants of the Cauchy-like matrix $T_Q A_m T_Q^T$. In other words, $T_Q \cdot C_Q(A_m) \cdot T_Q^T$ is simply a diagonal part of $T_Q A_m T_Q^T$, implying (10.5).

The above arguments indicate that there is a close connection between finding an optimal Frobenius-norm approximant of a Toeplitz matrix, and transformations of Toeplitz matrices to Cauchy-like matrices. Such transformations were closely studied in several recent papers. For example, in [O93b], [O95] and [He95b] direct (i.e., based on the computation on the matrix entries, without explicit use of displacement operators) transformations were considered. Their results can be applied to obtain T.Chan-type preconditioners for the transforms II and III.

In this paper explicit formulas are obtained not only for T.Chan-type, but also for G.Strang-type preconditioners. Moreover, in obtaining T.Chan-type preconditioners we follow [GO94a], [GKO95], [KO95a], [KO94], [KO95b]), and explore a different approach to transformations to Cauchy-like matrices. The crucial point here is to introduce an appropriate displacement operator, ∇_{H_Q}, where H_Q is diagonalized by a unitary matrix. New transformation formulas (systematically obtained here for all 8 cases) require only one discrete trigonometric transform to compute the diagonal part of a Cauchy-like matrices, as compared to 2 such transforms in [O93b], [O95], [He95b]. Furthermore, the concept of partially reconstructible matrices suggested to use the definition of ∇_{H_Q}-generator, given in [KO95a]. This allowed us to obtain a unified descriptions for the both G.Strang-type and T.Chan-type preconditioners, given in Tables 10 and 9, resp. These new formulas allowed us to establish in Sec. 9 the result on close assimptotic behavior of both classes of preconditioners, and to prove the crucial clustering property for $C_Q(A_m)$.

11 Real-arithmetic algorithms for multiplication of a Toeplitz matrix by a vector

11.1. Real symmetric Toeplitz matrices. In the first part of the paper we developed two families of G.Strang-type and of T.Chan-type preconditioners for real symmetric Toeplitz matrices, and established the properties 1-3, guaranteeing a convergence for the PCGM (under the Wiener class assumption). In this and the next sections we address the question of how to efficiently organize the iteration process itself.

First observe that all the computations with new preconditioners (i.e., their construction, and then solving the associated linear systems) can be done in real arithmetic. To fully exploit this advantageous property we have to specify an efficient real-arithmetic algorithm for multiplication of a Toeplitz matrix by a vector (the standard technique is based on the FFT, assuming complex arithmetic). In this section we observe that the explicit formulas obtained for the G.Strang-type readily suggest such algorithms for all 8 cases. These algorithms can be derived in two following steps.

- First, we embed a $m \times m$ Toeplitz matrix A_m into a larger $2m \times 2m$ Toeplitz matrix $\mathcal{A}_{2m}$ by padding its first column with m zeros.
- Secondly, we construct for $\mathcal{A}_{2m}$ the G.Strang-type preconditioner $S_Q(\mathcal{A}_{2m})$.

As was shown in Sec. 7, this preconditioner admits a Toeplitz-plus-Hankel-plus-border representation,

$$S_Q(\mathcal{A}_{2m}) = \mathcal{A}_{2m} + \mathcal{H} + \mathcal{B},$$

with the Hankel part $\mathcal{H}$ and the "border" part $\mathcal{B}$ displayed in Table 9. Now taking into account the banded structure of $\mathcal{A}_{2m}$, one sees that in all 8 cases the Hankel and the "border" part do not affect the central part of $S_Q(\mathcal{A}_{2m})$. In other words, our initial matrix A_m is a submatrix of $S_Q(\mathcal{A}_{2m})$. This observation allows us to use any of 8 DCT's or DST's to multiply a real symmetric Toeplitz matrices by a vector in only two discrete trigonometric transforms of the order $2m$ (one more such transform is needed to compute the diagonal form for $S_Q(A)$). For the case DST-I such an algorithm was proposed earlier by Boman and Koltracht in [BK95].

Although this is beyong our needs in the present paper, note that the formulas for the G.Strang-type preconditioners allow us to multiply in the same way Toeplitz-plus-Hankel matrices by a vector. These algorithms are analogs of the well-known embedding-into-circulant (complex arithmetic) multiplication algorithm. There is another well-known (complex arithmetic) method for multiplying a Toeplitz matrix A_m by a vector, based on the decomposition of A_m into a sum of circulant and skew-circulant matrices. Real-arithmetic discrete-trigonometric transform based analogs of this algorithm are offered in Sec. 12. However, before presenting them it is worth to note that the formulas for the G.Strang-type preconditioners also allow a multiplication of a non-symmetric Toeplitz by a vector.

11.2. Non-symmetric Toeplitz matrices. Clearly, a multiplication of a Toeplitz matrix by a vector is trivially reduced to two such multiplication for its lower and upper triangular parts.

These problems can be solved via exactly the same embedding technique used in the previous subsection. Indeed, the lower left $m \times m$ block of the matrix $S_Q(\mathcal{A}_{2m})$ is an upper triangular Toeplitz matrix, an observation immediately leading to the desired set of algorithms.

11.3. Polynomial interpretation. In fact, the latter algorithms of Sec. 11.2 were obtained earlier in [O93a], which was made available to several experts in the field during various times. However the arguments of [O93a] were polynomial, and it is instructive to include this useful interpretation.

1). • First, let us consider two polynomials

$$r(x) = r_0 Q_0(x) + r_1 Q_1(x) + \cdots + r_{2m-2} Q_{2m-2}(x)$$

$$s(x) = b_0 Q_0(x) + b_1 Q_1(x) + \cdots + b_{m-1} Q_{m-1}(x)$$

where $\{r_k\}$ are computed by

$$\begin{bmatrix} r_0 \\ \vdots \\ r_{m-1} \end{bmatrix} = G_Q \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

with G_Q being the corresponding matrix of Table 10.

• Let us evaluate $r(x)$ and $s(x)$, respectively, at the zeros of Q_{2m-1} by computing

$$V_Q \cdot \begin{bmatrix} r_0 \\ \vdots \\ r_{2m-1} \end{bmatrix}, \qquad V_Q \cdot \begin{bmatrix} b_0 \\ \vdots \\ b_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Using the results in Table 3 it can be done in just two discrete trigonometric transforms (associated with Q) of the order $2m$.

2). By multiplying the obtained numbers we compute the values of $r(x)s(x)$ at zeros of $Q_{2m-1}(x)$.

3). Finally the coefficients of

$$r(x)s(x) = p_0 Q_0(x) + p_1 Q_1(x) + \cdots + p_{2m-2} Q_{2m-2}(x) \tag{11.1}$$

are recovered via one more (inverse) discrete trigonometric transform of the order $2m$.

The latter real-arithmetic procedure is clearly efficient, however the question is how the obtained numbers $\{p_k\}$ are related to our purpose, i.e., to the convolution of $\{a_k\}$ and $\{b_k\}$. By using the properties of the corresponding polynomials $\{Q_k(x)\}$ one can show that in each of the 8 cases the last m coefficients of $r(x)s(x)$ in (11.1) give us the desired convolution of $\{a_{m-1}, \ldots, a_0\}$ and $\{b_{m-1}, \ldots, b_0\}$. For example, in cases of DCT-I and DCT-III it follows from the following property of Chebyshev polynomials

$$2T_k(x)T_j(x) = T_{k+j}(x) + T_{|k-j|}(x).$$

However, a matrix interpretation of these algorithms reduces to

$$S_Q(\mathcal{A}_{2m}) \cdot b = T_Q^T \cdot (T_Q S_Q(\mathcal{A}_{2m}) T_Q^T) \cdot (T_Q b),$$

which with the results in Sec. 11.2 and Table 7 make it clear that in all 8 cases the last coefficients in (11.1) will give us the desired convolution.

12 Transformation-to-Vandermonde-like approach for the PCGM

12.1. Transformations. As was detailed in Sec. 4, polynomial Hankel-like matrices (this class includes Toeplitz-like matrices) can be transformed into Cauchy-like matrices. In several recent papers this idea has been found to be useful for design of accurate *direct* methods for solving Toeplitz linear systems.

In this section we suggests an application of this technique to PCGM for Toeplitz matrices. More specifically, instead of applying PCGM to the preconditioned system

$$S_Q(A_m)^{-1} A_m x = b, \tag{12.1}$$

we suggest to apply it to the *transformed system*

$$(T_Q S_Q(A_m)^{-1} T_Q^T) \cdot (T_Q A_m) x = T_Q b$$

where the preconditioner is transformed to the diagonal matrix $(T_Q S_Q(A_m)^{-1} T_Q^T)$, and the Toeplitz matrix A_m is transformed into a Vandermonde-like matrix $T_Q A_m$. Since a diagonal linear system can be solved in m operations, such transformation saves us 2 discrete transforms per iteration, if we can multiply the Vandermonde-like matrix $T_Q A_m$ by a vector with exactly the same complexity as the initial matrix A_m. In the rest of the section we describe algorithms for this purpose, these new algorithms are based on the formulas, which are counterparts of the well-know decomposition of a Toeplitz matrix into a sum of a circulant and a skew-circulant matrices.

12.2. Discrete transforms II and IV. For example, from the Toeplitz-plus-Hankel-plus-border decompositions in Table 7 it immediately follows that ,

$$A_m = \frac{1}{2}(S_{C2}(A_m) + S_{S2}(A_m)), \qquad A_m = \frac{1}{2}(S_{C4}(A_m) + S_{S4}(A_m)). \tag{12.2}$$

where we denote by $C1, C2, C3, C4, S1, S2, S3, S4$ the corresponding polynomial systems Q of Table 2.Since the preconditioners $S_Q(A_m)$ are diagonalized by the corresponding transform matrices T_Q, each of these formulas clearly allow us to multiply A_m by a vector in just 4 discrete trigonometric transforms (with 2 more transforms needed only once to diagonalize $S_Q(A)$). As to our goal, i.e., $T_Q A_m$, we have the following formulas.

Table 12. Decompositions for II and IV transforms.

DCT-II	$T_{C2}A_m = D_{C2}T_{C2} + T_{C2}T_{S2}^T D_{S2}T_{S2}$	DST-II	$T_{S2}A_m = T_{S2}T_{C2}^T D_{C_2}T_{C2} + D_{S2}T_{S_2}$
DCT-IV	$T_{C4}A_m = D_{C4}T_{C4} + T_{C4}T_{S4}^T D_{S4}T_{S4}$	DST-IV	$T_{S4}A_m = T_{S4}T_{C4}^T D_{C_4}T_{C4} + D_{S4}T_{S_4}$

Here

$$D_Q = W_Q T_Q R_Q \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix} \tag{12.3}$$

where the matrices R_Q are displayed in Table 10. These formulas reduce the complexity of one iteration to 4 real discrete trigonometric transform of the order m, as compared to 6 such transforms of the methods in the previous section.

12.3. Discrete transforms I. Thus for the II and IV transforms the formulas (12.2) seem to be simple because in these cases the Hankel part of the corresponding cosine and sine G.Strang-type preconditioners differ only by the sign (see, e.g., Table 10). For the I and III transforms this is not so, and the reason seem to be that the definitions of the corresponding discrete transforms are not chosen to imply for them the representations of the form (12.2). However, instead of changing the standard definitions (for example, taking care of different $N+1$ and $N-1$ and of the size for the DCT-I, DST-I, DCT-III and DST-III), we show that even with standard definitions in the remaining two cases one can derive not much more involved formulas, also leading to the same efficiency of 4 discrete transforms per iteration.

Indeed, in the case of DCT-I and DST-I we have the following. Let the numbers $\{c_k\}$, $\{s_k\}$ be defined by

$$(I+(Z^T)^2)\begin{bmatrix} f_0 \\ \vdots \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix}, \qquad \begin{bmatrix} e_0 \\ \vdots \\ e_{n-1} \end{bmatrix} = (Z^T)^2 \begin{bmatrix} f_0 \\ \vdots \\ f_{n-1} \end{bmatrix}$$

where Z denotes the lower shift matrix. Then clearly

$$A = S_{C1}(E_m) + S_{S1}(F_m) - B_{C1}$$

where $S_{C1}(E_m)$, $S_{S1}(F_m)$ are G.Strang-type preconditioners from Table 7 for Toeplitz matrices E_m and F_m defined by their first columns $[e_k]$ and $[f_k]$, resp. The matrix B_{C1} is the border matrix of F_m defined in the row DCT-I of the same Table. Therefore we have the following formulas.

Table 12. Continuation.

DCT-I	$T_{C1}A_m = D_{C1}T_{C1} + T_{C1}(T_{S1}^T D_{S1} T_{S1} + B_{C1})$
DST-I	$T_{S1}A_m = T_{S1}(T_{C1}^T D_{C1} T_{C1} + B_{C1}) + D_{S1}T_{S1}$

Here all the diagonal matrices are obtained by (12.3) with the replacement of $[a_k]$ by the $[f_k]$ and $[e_k]$, resp. Since B_{C1} is the rank-four matrix, these formulas allow us to compute the product of $T_Q A_m$ by a vector in 4 real trigonometric transforms of the order m. Note that a different formula of this kind was obtained for DST-I in [H95].

12.4. Discrete transforms III. In this case we have

$$A_m = \frac{1}{2}(S_{C3}(A_m) - B_{C3} + Z S_{S3}(A_m) Z^T),$$

$$A_m = \frac{1}{2}(Z^T S_{C3}(A_m) Z + S_{S3}(A_m) - B_{S3}),$$

leading to the formulas in the next Table.

Table 12. Continuation.

DCT-III	$T_{C3}A_m = \frac{1}{2}(D_{C3}T_{C3} + T_{C3}(Z T_{S3}^T D_{S3} T_{S3} Z^T - B_{C3})$
DST-III	$T_{S3}A_m = \frac{1}{2}(T_{S1}(Z^T T_{C3}^T D_{C3} T_{C3} Z - B_{S3}) + D_{S3}T_{S3}$

Again the complexity of one iteration is 4 real discrete trigonometric transforms per iteration.

13 Suggestion on how to speed-up numerical experiments

Our first numerical experiments confirm the theoretical results obtained in the present paper. We believe that more research has to be done to theoretically and numerically compare new preconditioners with the other existing ones, and to make reliable recommendations depending upon particular classes of Toeplitz matrices and their generating functions. Testing the clustering property for a preconditioner P requires computing the number of eigenvalues of $P^{-1}A_m$ outside of an interval $[1-\varepsilon, 1+\varepsilon]$. This can be done by checking the inertia for (in genral indefinite) two matrices

$$P^{\frac{1}{2}}A_mP^{\frac{1}{2}} - (1+\varepsilon)I, \qquad P^{\frac{1}{2}}A_mP^{\frac{1}{2}} - (1-\varepsilon)I, \tag{13.1}$$

The use of the standard methods requires $O(n^3)$ operations, which makes difficulties when performing numerical tests for large matrices. It was observed in [TS96] that if P is circulant, then this disadvantage can be overcome, because the Toeplitz-like structure of the two matrices in (13.1) allows one to compute for them the LDL^* factorization in only $O(n^2)$ operations.

We propose an alternative method, based on the transformation-to-Cauchy-like approach. We suggest to consult [KO95a] for more details, and here only briefly clarify the main points. More precisely, for our G.Strang-type and T.Chan-type preconditioners we propose to check the inertia for the (congruent) Cauchy-like matrix

$$R = (T_Q P^{\frac{1}{2}} T_Q^T) \cdot (T_Q A_m T_Q^T) \cdot (T_Q P^{\frac{1}{2}} T_Q^T) - \mu I, \tag{13.2}$$

as follows.

- First, we represent a Toeplitz matrix A_m by its ∇_{H_Q}-generator $\{G_Q, J, C_Q(A_m)\}$, see, e.g., Sec. 10.2.
- Then A_m can be transformed to a Cauchy-like matrix $T_Q(A_m)T_Q^T$; as indicated in Sec. 10.2, this transformation can be done in just 3 discrete trigonometric transforms computing the Cauchy-like generator for the latter matrix (i.e., avoiding any operations on its entries.) In all cases the corresponding displacement ranks do not exceed 4.
- Multiplying the obtained Cauchy-like matrix by diagonal matrices $(T_Q P^{\frac{1}{2}} T_Q^T)$ from both sides and subtracting μI from its kernel part, we obtain the desired Cauchy-like matrix R in (13.2). Again this operation can done by avoiding manipulation by the matrix entries, and by just scaling the corresponding generator.
- Finally, the inertia of R can be checked in $O(n^2)$ operations in a numerically reliable manner by applying to R the fast Gaussian elimination with Bunch-Kaufman pivoting, designed for partially reconstructible Cauchy-like matrices in [KO95a].

References

[BB90] D.Bini and F. Di Benedetto, *A new preconditioner for the parallel solution of of positive definite Toeplitz systems*, in Proc. 2nd ACM Symp. on Parallel Algorithms and Architectures, Crete, Greece, 1990, 220-223.

[BC83] D. Bini and M.Capovani, *Spectral and computational properties of band symmetric Toeplitz matrices*, Linear Algebra Appl., **52** (1983), 99-126.

[BK95] E.Boman and I.Koltracht, *Fast transform based preconditioners for Toeplitz equations*, SIAM J. on Matrix Analysis and Appl., **16**(1995), 628-645.

[BKO94] T.Boros, T.Kailath and V.Olshevsky, *Predictive partial pivoting and backward stability of fast Cauchy solvers*, 1994, preprint.

[C88] T.Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Stat. Comput., **9**(1988), 766-771.

[C89] R.Chan, *Circulant preconditioners for Hermitioan Toeplitz systems*, SIAM J. Matrix Analysis and Appl., **10**(1989), 542-550.

[CJY91] R.Chan, X.Jin and M.Yeung, *The circulant operator in the Banach algebra of matrices*, Linear Algebra Appl., **149** (1991), 41-53.

[CNP94] R.Chan, J.Nagy and R.Plemmons, *Displacement preconditioner for Toeplitz least squares iteration*, Electronic Transactions on Numerical Analysis, **2**(1994), 44-56.

[CS89] R.Chan and G.Strang, *Toeplitz equations by conjugate gradients with circulant preconditioner*, SIAM J. Sci. Stat. Comp., **10** (1989), 104-119.

[CY92] R.Chan and M.Yeng, *Circulant preconditioners for Toeplitz matrices with positive continuous generating functions*, Math. Comp., **58**(1992), 233-240.

[CN96] R.Chan and K.Ng, *Conjugate gradient method for Toeplitz systems*, SIAM Review, **38**(1996), 427 -482.

[DFZ95] C. Di Fiore and P.Zellini, *Matrix decompositions using displacement rank and classes of commutative matrix algebras*, Linear Algebra Appl. **229** (1995), 49-99.

[GK89] I.Gohberg and I.Koltracht, *Efficient algorithm for Toeplitz plus Hankel matrices*, Integral Equations and Operator Theory, **12** (1989), 136 – 142.

[GKO95] I.Gohberg, T.Kailath and V.Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. of Computation, **64** (1995), 1557-1576.

[GL89] G. Golub and C, Van Loan, *Matrix Computations*, second edition, Johns Hopkins U. P., Baltimore, 1989.

[GO94a] I.Gohberg and V.Olshevsky, *Complexity of multiplication with vectors for structured matrices*, Linear Algebra Appl., **202** (1994), 163 – 192.

[GO94b] I.Gohberg and V.Olshevsky, *Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems*, Integral Equations and Operator Theory, **20**, No. 1, 1994, 44 – 83.

[GS84] U.Grenader and G.Szegö, *Toeplitz forms and their applications*, 2nd ed., Chelsea, New York, 1984.

[Gu95] Ming Gu, *Stable and efficient algorithms for structured systems of linear equations*, preprint, 1995.

[Gu96] Ming Gu, *Fast and Stable algorithms for Toeplitz linear equations*, preprint, 1996.

[He95a] G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, in : Linear Algebra in Signal Processing, IMA volumes in Mathematics and its Applications, vol. **69** (1995), 95 - 114.

[He95b] G.Heinig, *Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices. I. Transformations.* to appear in Linear Algebra Appl., Proc. of ILAS-95.

[H95] T.Huckle, *Cauchy matrices and iterative methods for Toeplitz matrices*, in Proc. of SPIE-95, **2563**(1995), 281-292.

[HJR88] G.Heinig, P.Jankowski and K.Rost, *Fast inversion of Toeplitz-plus-Hankel matrices*, Numer.Math. **52** (1988), 665 – 682.

[K87] T.Kailath, *Signal processing applications of some moment problems*, Proc. of Symposia in Appl. Math., vol. **37**, 71-109. AMS annual meeting, short course reprinted in *Moments in mathematics*, ed. H.Landau, San Antonio, TX, January 1987.

[KKM79] T.Kailath, S.Kung and M.Morf, *Displacement ranks of matrices and linear equations*, J. Math. Anal. and Appl., **68** (1979), 395-407.

[KO94] T.Kailath and V.Olshevsky, *Displacement structure approach to polynomial Vandermonde and related matrices*, preprint, 1994.

[KO95a] T.Kailath and V.Olshevsky, *Bunch-Kaufman Pivoting for Partially Reconstructable Cauchy-like Matrices, with Applications to Toeplitz-like Linear Equations and to Boundary Rational Matrix Interpolation Problems*, to appear Linear Algebra and Appl., (1997), Proc. of the 5-th ILAS conference, 1995.

[KO95b] T.Kailath and V.Olshevsky, *Displacement structure approach to Chebyshev-Vandermonde and related matrices*, Integral equations and Operator Theory, **22**, 1995, 65-92.

[MB79] J.Maroulas and S.Barnett, Polynomials with respect to a general basis. I. Theory, *J. of Math. Analysis and Appl.*, **72** : 177 -194 (1979).

[O93a] V.Olshevsky, *A fast real-arithmetic algorithm for multiplication of a Toeplitz matrix by a vector*, preprint, 1993.

[O93b] M. Ohsmann, *Fast cosine transform of Toeplitz matrices, algorithm and applications*, IEEE Transactions on Signal Processing, **41, No. 10** (1993), 3057-3061.

[O95] M.Ohsmann, *Fast transforms of Toeplitz matrices*, Linear Algebra Appl., **231** (1995), 181-192.

[P90] V.Pan, *On computations with dense structured matrices*, Math. of Computation, **55**, No. 191 (1990), 179 – 190.

[S86] G.Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., **74** (1986), 171-176.

[SB95] D.Sweet and R.Brent, *Error analysis od a partial pivoting method for structured matrices*, Advanced Signal processing algorithms, Proc of SPIE-1995, vol. **2563**, 266-280.

[T92] E.Tyrtyshnikov, *Optimal and superoptimal circulant preconditioners*, SIAM J. on Matrix Analysis Appl., **13**(1992), 459-473.

[T95] E.Tyrtyshnikov, *Circulant preconditioners with unbounded inverses*, Linear Algebra Appl., **216** (1995), 1-24.

[TS96] E.Tyrtyshnikov and V.Strela, *Which preconditioner is better?*, Math. Comp., **65**(1996), 137-150.