

BLOCK DIAGONAL AND SCHUR COMPLEMENT PRECONDITIONERS FOR BLOCK TOEPLITZ SYSTEMS WITH SMALL SIZE BLOCKS

WAI-KI CHING*, MICHAEL K. NG†, AND YOU-WEI WEN‡

Abstract. In this paper we consider the solution of Hermitian positive definite block-Toeplitz systems with small size blocks. We propose and study block diagonal and Schur complement preconditioners for such block-Toeplitz matrices. We show that for some block-Toeplitz matrices, the spectra of the preconditioned matrices are uniformly bounded except for a fixed number of outliers and this fixed number depends only on the size of block. Hence conjugate gradient type methods, when applied to solving these preconditioned block-Toeplitz systems with small size blocks, converge very fast. Recursive computation of such block diagonal and Schur complement preconditioners are considered by using the nice matrix representation of the inverse of a block-Toeplitz matrix. Applications to block-Toeplitz systems arising from least squares filtering problems and queueing networks are presented. Numerical examples are given to demonstrate the effectiveness of the proposed method.

AMS subject classifications. 65F10, 65N20.

Key words. Block-Toeplitz matrix, block diagonal, Schur complement, preconditioners, recursion

1. Introduction. In this paper we consider the solution of a Hermitian positive definite block-Toeplitz (BT) system with small size blocks

$$A_{n,m}X = B \tag{1.1}$$

where X and B are mn -by- m matrices and

$$A_{n,m} = \begin{pmatrix} A_0 & A_{-1} & \cdots & A_{1-n} \\ A_1 & A_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_{-1} \\ A_{n-1} & \cdots & A_1 & A_0 \end{pmatrix}$$

where each A_j is an m -by- m matrix with $A_j = A_{-j}^*$ and m is much smaller than n . Here “*” denotes the conjugate transpose. This kind of linear systems arises from many applications such as the multichannel least squares filtering in time series [33], signal and image processing [27] and queueing system [15]. We will discuss these applications, in particular the least squares filtering problems and queueing networks in Section 5.

Recent research on using the preconditioned conjugate gradient method as an iterative method for solving n -by- n Toeplitz systems has brought much attention. One

*Advanced Modeling and Applied Computing Laboratory and Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: wkc@maths.hku.hk. The research of this author is supported in part by Hong Kong Research Grants Council Grant No. HKU 7126/02P and HKU CRCG Grant Nos. 10204436, 10205105.

†Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: mng@math.hkbu.edu.hk. The research of this author is supported in part by Hong Kong Research Grants Council Grant Nos. 7046/03P, 7035/04P, 7035/05P and HKBU FRGs.

‡Advanced Modeling and Applied Computing Laboratory and Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong. The research of this author is supported in part by the grant of the project of oil resources in pre-cenozoic layer in Gulf of Bohai No. KZCX1SW-18, and Guangdong Natural Science Foundation of Grant No. 32475.

of the main important results of this methodology is that the complexity of solving a large class of Toeplitz systems can be reduced to $O(n \log n)$ operations provided that a suitable preconditioner is chosen under certain conditions on the Toeplitz matrix [10]. Circulant preconditioners [5, 6, 11, 13, 14, 24, 32, 37, 40], banded-Toeplitz preconditioners [7] and multigrid methods [8, 17] have been proposed and analyzed. In these papers, the diagonals of the Toeplitz matrix are assumed to be the Fourier coefficients of a certain generating function.

In the literature, there are some papers [25, 28, 29, 34, 35, 36, 38] discussing about iterative block-Toeplitz solvers. In [28, 35, 36], they considered n -by- n block Toeplitz matrices with m -by- m blocks generated by a Hermitian matrix-valued generating function and analyzed the associated problem of preconditioning using preconditioners generated a nonnegative definite, not essentially singular, matrix-valued functions. In [25, 29, 34], they considered block-Toeplitz-Toeplitz-block matrices and studied block band-Toeplitz preconditioners. In [38], multigrid methods were applied to solving block-Toeplitz-Toeplitz-block systems. In the above methods, the underlying generating functions are assumed to be known in order to construct the preconditioners.

In this paper, we also consider block-Toeplitz matrices $A_{n,m}$ generated by a matrix-valued function:

$$F_m(\theta) = [f_{u,v}(\theta)]_{1 \leq u,v \leq m},$$

where $f_{u,v}(\theta)$ are 2π -periodic functions. Under this assumption, the block A_j of $A_{n,m}$ is given by

$$A_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_m(\theta) e^{-ij\theta} d\theta.$$

When $F_m(\theta)$ is nonnegative definite and is not essentially singular, the associated block-Toeplitz matrix $A_{n,m}$ is positive definite [28, 35]. For such block-Toeplitz matrices, Serra [35] has investigated block-Toeplitz preconditioners and studied the spectral property of these preconditioned matrices. He proved that if the block-Toeplitz preconditioner is generated by $G_m(\theta)$, the generalized Rayleigh quotient, related to matrix functions $F_m(\theta)$ and $G_m(\theta)$ is contained in a set of the form (c_1, c_2) with $0 < c_1$ and $c_2 < \infty$, then the preconditioned conjugate gradient (PCG) method requires only a constant number of iterations in order to solve, within a preassigned accuracy, the given block-Toeplitz system.

In [31], Ng et al. have proposed to use recursive-based PCG methods for solving Toeplitz systems. The idea is to use a principal submatrix of a Toeplitz matrix as a preconditioner. The inverse of the preconditioner can be constructed recursively by using the Gohberg-Semencul formula. They have shown that this method is competitive with the method of circulant preconditioners. In this paper, we study block diagonal and Schur complement preconditioners for block-Toeplitz systems. We note that there is a natural partitioning of the block-Toeplitz matrix in 2-by-2 blocks as follows:

$$A_{n,m} = \begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ A^{(2,1)} & A^{(2,2)} \end{pmatrix}. \quad (1.2)$$

Here $A^{(1,1)}$ and $A^{(2,2)}$ are the principal submatrices of $A_{n,m}$. They are also block-Toeplitz matrices generated by the same generating function of $A_{n,m}$. Therefore it is

natural and important to examine if the corresponding system

$$\begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ A^{(2,1)} & A^{(2,2)} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \quad (1.3)$$

can be solved efficiently by exploiting this partitioning. Here we consider preconditioning $A_{n,m}$ by a block diagonal matrix

$$B_{n,m} = \begin{pmatrix} A^{(1,1)} & 0 \\ 0 & A^{(2,2)} \end{pmatrix}.$$

Since both $A^{(1,1)}$ and $A^{(2,2)}$ are block-Toeplitz matrices generated by the same generating function $F_m(\theta)$, we particularly consider $B_{n,m}$ in the following form:

$$B_{n,m} = \begin{pmatrix} A_{n/2,m} & 0 \\ 0 & A_{n/2,m} \end{pmatrix}.$$

Here without loss of generality, we may assume n is even. We note that if $A_{n,m}$ is positive definite, then $B_{n,m}$ is also positive definite and the eigenvalues of the preconditioned matrix $B_{n,m}^{-1}A_{n,m}$ lie in the interval $(0, 2)$.

On the other hand, the Schur complement arises when we use a block factorization of (1.2). The linear system (1.3) becomes

$$\begin{pmatrix} I & 0 \\ A^{(2,1)}(A^{(1,1)})^{-1} & I \end{pmatrix} \begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ 0 & S_{n,m} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \quad (1.4)$$

where

$$S_{n,m} = A^{(2,2)} - A^{(2,1)}(A^{(1,1)})^{-1}A^{(1,2)}.$$

We see that the method requires the formation of the Schur complement matrix. Therefore we consider approximating $S_{n,m}$ by $A^{(2,2)} = A_{n/2,m}$ and study the preconditioner of the form:

$$\begin{aligned} C_{n,m} &= \begin{pmatrix} I & 0 \\ A^{(2,1)}(A^{(1,1)})^{-1} & I \end{pmatrix} \begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ 0 & A^{(2,2)} \end{pmatrix} \\ &= \begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ A^{(2,1)} & A^{(2,2)} + A^{(2,1)}(A^{(1,1)})^{-1}A^{(1,2)} \end{pmatrix}. \end{aligned} \quad (1.5)$$

We note that if $A_{n,m}$ is positive definite, then $C_{n,m}$ is also positive definite and the eigenvalues of the preconditioned matrix $C_{n,m}^{-1}A_{n,m}$ are inside of the interval $(0, 1]$. In particular, there are at least $mn/2$ eigenvalues of the preconditioned matrix being equal to one.

The main results of this paper is that if the generating function $F_m(\theta)$ is Hermitian positive definite, and is spectrally equivalent to

$$G_m(\theta) = [g_{u,v}]_{1 \leq u, v \leq m}$$

where $g_{u,v}$ are trigonometric polynomials, then the spectra of the preconditioned matrices $B_{n,m}^{-1}A_{n,m}$ and $C_{n,m}^{-1}A_{n,m}$ are uniformly bounded except for a fixed number of outliers where the number of outliers depends only on m . Hence the conjugate gradient type methods, when applied to solving these preconditioned block-Toeplitz systems, converge very quickly, especially when m is small.

We remark the construction of our preconditioners does not require the underlying matrix generating functions. However, the inverse of block-Toeplitz matrix $A^{(1,1)}$ is required. Using the same idea in [31], we employ Gohberg-Semencul formula to represent the form of the inverse of $A^{(1,1)}$, and apply a recursive method to construct the inverse of $A^{(1,1)}$. It is important to note that we do not directly use the Gohberg-Semencul formula to generate the solution of the original block-Toeplitz system. We remark that the solution results are not accurate when the block-Toeplitz matrices are ill-conditioned. Indeed we use the Gohberg-Semencul formula to generate a preconditioner, and then use the preconditioned conjugate gradient method with this preconditioner to compute the solution of the original system iteratively.

The outline of this paper is as follows. In §2, we analyze the spectra of the preconditioned matrices. In §3, we describe the recursive algorithms for block diagonal and Schur complement preconditioners. Numerical results are given in §4 to illustrate the effectiveness of our approach. Finally, concluding remarks are given in §5 to address further research issues.

2. Analysis of Preconditioners. In this section, we analyze the spectra of the preconditioned matrices $B_{m,n}^{-1}A_{n,m}$ and $C_{n,m}^{-1}A_{n,m}$.

We first note that since $A_{n,m}$ is positive definite, we have the following results given in [1, pp.374-377].

LEMMA 2.1. *Let \mathbf{x} and \mathbf{y} be $mn/2$ -vector. Define*

$$\gamma = \sup_{\mathbf{x}, \mathbf{y}} \frac{\mathbf{x}^* A_{n/2, m}^{(1)} \mathbf{y}}{\sqrt{\mathbf{x}^* A_{n/2, m} \mathbf{x}_1 \cdot \mathbf{y}^* A_{n/2, m} \mathbf{y}}}.$$

If $A_{n,m}$ is Hermitian and positive definite, then $\gamma < 1$. In particular, we have

$$\gamma^2 = \sup_{\mathbf{y}} \frac{\mathbf{y}^* A_{n/2, m}^{2,1} A_{n/2, m}^{-1} A_{n/2, m}^{1,2} \mathbf{y}}{\mathbf{y}^* A_{n/2, m} \mathbf{y}}.$$

Using Lemma 2.1 and the assumption that $A_{n,m}$ is Hermitian and positive definite, we have the following results.

- The eigenvalues of the preconditioned matrix $B_{n,m}^{-1}A_{n,m}$ lie inside the interval $(0, 2)$. Also if μ is an eigenvalue of $B_{n,m}^{-1}A_{n,m}$, then $2 - \mu$ is also an eigenvalue of $B_{n,m}^{-1}A_{n,m}$.
- The eigenvalues of $C_{n,m}^{-1}A_{n,m}$ are inside the interval $(0, 1]$. Moreover, at least $mn/2$ eigenvalues of $C_{n,m}^{-1}A_{n,m}$ are equal to 1.

We then show that the eigenvalues of $B_{n,m}^{-1}A_{n,m}$ and $C_{n,m}^{-1}A_{n,m}$ are uniformly bounded except for a fixed number of outliers for some generation functions $F_m(\theta)$. We first let

$$E_n(\theta) = [e_{u,v}(\theta)]_{1 \leq u, v \leq n} \quad \text{where} \quad e_{u,v}(\theta) = e^{-i(u-v)\theta}.$$

The block-Toeplitz matrix $A_{n,m}$ can be expressed in terms of its generating function:

$$A_{n,m} = \frac{1}{2\pi} \int_{-\pi}^{\pi} E_n(\theta) \otimes F_m(\theta) d\theta. \quad (2.1)$$

Similarly, the block diagonal preconditioner can be expressed as follows:

$$B_{n,m} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \begin{pmatrix} E_{n/2}(\theta) & 0 \\ 0 & E_{n/2}(\theta) \end{pmatrix} \otimes F_m(\theta) d\theta. \quad (2.2)$$

We note that there exists a permutation matrix $P_{n,m}$ such that

$$P_{n,m}^t A_{n,m} P_{n,m} = \tilde{A}_{n,m} = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_m(\theta) \otimes E_n(\theta) d\theta$$

and

$$P_{n,m}^t B_{n,m} P_{n,m} = \tilde{B}_{n,m} = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_m(\theta) \otimes \begin{pmatrix} E_{n/2}(\theta) & 0 \\ 0 & E_{n/2}(\theta) \end{pmatrix} d\theta.$$

It is clear that $\tilde{A}_{n,m}$ and $\tilde{B}_{n,m}$ are Toeplitz-block (TB) matrices, and the spectra of $A_{n,m}$ and $\tilde{A}_{n,m}$, and $B_{n,m}$ and $\tilde{B}_{n,m}$ are the same. Since the spectra of $B_{n,m}^{-1} A_{n,m}$ and $\tilde{B}_{n,m}^{-1} \tilde{A}_{n,m}$ are the same, it suffices to study the spectral properties of $\tilde{B}_{n,m}^{-1} \tilde{A}_{n,m}$.

We give the following two lemmas.

LEMMA 2.2. *Let $A = [a_{i,j}]_{1 \leq i,j \leq n}$ and $B = [b_{i,j}]_{1 \leq i,j \leq m}$. Then for any m -by- n matrices $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ and $Y = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)$, we have*

$$\text{vec}(X)^*(A \otimes B)\text{vec}(Y) = \sum_{u=1}^n \sum_{v=1}^n a_{u,v} \mathbf{x}_u^* B \mathbf{y}_v \quad (2.3)$$

where

$$\text{vec}(X) = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} \quad \text{and} \quad \text{vec}(Y) = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_n \end{pmatrix}.$$

LEMMA 2.3. *Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$ with*

$$\mathbf{x}_l = (x_{(l-1)n+1}, x_{(l-1)n+2}, \dots, x_{ln})^t \quad (1 \leq l \leq m),$$

and

$$\mathbf{p}_1(\theta) = (\check{\mathbf{p}}_{11}(\theta), \check{\mathbf{p}}_{21}(\theta), \dots, \check{\mathbf{p}}_{m1}(\theta))^t \quad \text{and} \quad \mathbf{p}_2(\theta) = (\check{\mathbf{p}}_{12}(\theta), \check{\mathbf{p}}_{22}(\theta), \dots, \check{\mathbf{p}}_{m2}(\theta))^t$$

with

$$\check{\mathbf{p}}_{j1}(\theta) = \sum_{l=1}^{n'} x_{(j-1)n+l} e^{-i(l-1)\theta} \quad \text{and} \quad \check{\mathbf{p}}_{j2}(\theta) = e^{-in'\theta} \sum_{l=1}^{n-n'} x_{(j-1)n+n'+l} e^{-i(l-1)\theta}.$$

If $A_{n,m}$ is generated by $F_m(\theta)$, then we have

$$\mathbf{x}^t \tilde{B}_{n,m} \mathbf{x} = \frac{1}{2\pi} \int_{-\pi}^{\pi} [\mathbf{p}_1(\theta)^t F_m(\theta) \overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta)^t F_m(\theta) \overline{\mathbf{p}_2(\theta)}] d\theta \quad (2.4)$$

and

$$\mathbf{x}^t \tilde{A}_{n,m} \mathbf{x} = \mathbf{x}^t \tilde{B}_{n,m} \mathbf{x} + \frac{1}{2\pi} \int_{-\pi}^{\pi} [\mathbf{p}_1(\theta)^t F_m(\theta) \overline{\mathbf{p}_2(\theta)} + \mathbf{p}_2(\theta)^t F_m(\theta) \overline{\mathbf{p}_1(\theta)}] d\theta. \quad (2.5)$$

Proof. We construct $X = (\mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_m^t)$, i.e., $\mathbf{x} = \text{vec}(X)$. Using Lemma 2.2, we obtain

$$\text{vec}(X)^* \tilde{A}_{n,m} \text{vec}(X) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{u=1}^m \sum_{v=1}^m f_{u,v}(\theta) \mathbf{x}_u^* E_n(\theta) \mathbf{x}_v d\theta \quad (2.6)$$

and

$$\text{vec}(X)^* \tilde{B}_{n,m} \text{vec}(X) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{u=1}^m \sum_{v=1}^m f_{u,v}(\theta) \mathbf{x}_u^* \begin{pmatrix} E_{n/2}(\theta) & 0 \\ 0 & E_{n/2}(\theta) \end{pmatrix} \mathbf{x}_v d\theta. \quad (2.7)$$

We note that

$$\begin{aligned} & \mathbf{x}_u^* \begin{pmatrix} E_{n/2}(\theta) & 0 \\ 0 & E_{n/2}(\theta) \end{pmatrix} \mathbf{x}_v \\ &= \sum_{j=1}^{n/2} x_{(u-1)n+j} \sum_{l=1}^{n/2} x_{(v-1)n+l} e_{jl}(\theta) + \sum_{j=n/2+1}^n x_{(u-1)n+j} \sum_{l=n/2+1}^n x_{(v-1)n+l} e_{jl}(\theta) \\ &= \sum_{j=1}^{n/2} x_{(u-1)n+j} e^{-i(j-1)} \sum_{l=1}^{n/2} x_{(v-1)n+l} e^{i(l-1)} + \\ & \quad \sum_{j=n/2+1}^n x_{(u-1)n+j} e^{-i(j-1)} \sum_{l=n/2+1}^n x_{(v-1)n+l} e^{i(l-1)} \\ &= \check{\mathbf{p}}_{u1}(\theta) \overline{\check{\mathbf{p}}_{v1}(\theta)} + \check{\mathbf{p}}_{u2}(\theta) \overline{\check{\mathbf{p}}_{v2}(\theta)}. \end{aligned}$$

By using (2.7), one can obtain (2.4) directly. Similarly by using (2.6), (2.5) can also be derived. \square

Next we show that the eigenvalues of $B_{n,m}^{-1} A_{n,m}$ are uniformly bounded except for a fixed number of outliers when $F_m(\theta)$ is Hermitian positive definite, and is spectrally equivalent to $G_m(\theta) = [g_{u,v}]_{1 \leq u, v \leq m}$ where $g_{u,v}$ are trigonometric polynomials. We remark that the fixed number of outliers depends on m .

THEOREM 2.4. *Let $F_m(\theta)$ be Hermitian positive definite. Suppose $F_m(\theta)$ is spectrally equivalent to $G_m(\theta) = [g_{u,v}]_{1 \leq u, v \leq m}$ where $g_{u,v}$ are trigonometric polynomials and s is the largest degree of the polynomials in $G_m(\theta)$. Then there exist two positive numbers α and β ($\alpha < \beta$) independent of n such that for all $n > 2s'$ ($s' = \lceil s/2 \rceil$), at most $2ms'$ eigenvalues of $\tilde{B}_{n,m}^{-1} \tilde{A}_{n,m}$ (or $B_{n,m}^{-1} A_{n,m}$) are outside the interval $[\alpha, \beta]$.*

Proof. We note that there exist positive numbers γ_1 and γ_2 such that

$$0 < \gamma_1 \leq \frac{\mathbf{x}^* F_m(\theta) \mathbf{x}}{\mathbf{x}^* G_m(\theta) \mathbf{x}} \leq \gamma_2, \quad \forall \mathbf{x} \in \mathbf{R}^{mn}, \quad \forall \theta \in [0, 2\pi]. \quad (2.8)$$

We define the following two sets Υ and Ω as follows:

$$\Upsilon = \{r : r = jn + n/2 - s', jn + n/2 - s' + 1, \dots, jn + n/2 + s' - 1 \text{ for } j = 0, 1, \dots, m-1\}$$

and

$$\Omega = \{\mathbf{z} = (z_1, z_2, \dots, z_{mn})^t \mid z_k = 0 \text{ for } k \in \Upsilon\}.$$

We note that Ω is a $(mn - 2ms')$ -dimensional subspace in \mathbf{R}^{mn} . It follows that for $\mathbf{x} \in \Omega$ and $\mathbf{p}_u(\theta)$ ($u = 1, 2$) defined in Lemma 2.3, we have

$$\begin{aligned}
& \int_{-\pi}^{\pi} \mathbf{p}_1(\theta)^t G_m(\theta) \overline{\mathbf{p}_2(\theta)} d\theta = \int_{-\pi}^{\pi} \sum_{u=1}^m \sum_{v=1}^m \check{\mathbf{p}}_{u1}(\theta) f_{u,v}(\theta) \overline{\check{\mathbf{p}}_{v2}(\theta)} d\theta \\
& = \int_{-\pi}^{\pi} \sum_{u=1}^m \sum_{v=1}^m f_{u,v}(\theta) e^{in/2\theta} \sum_{j=1}^{n/2} x_{(u-1)n+j} e^{-i(j-1)\theta} \sum_{j=1}^{n/2} x_{(v-1)n+n/2+j} e^{i(j-1)\theta} d\theta \\
& = \sum_{u=1}^m \sum_{v=1}^m \int_{-\pi}^{\pi} f_{u,v}(\theta) e^{i(2s'+1)\theta} \sum_{j=1}^{n/2-s'} x_{(u-1)n+j} e^{-ij\theta} \cdot \\
& \quad \cdot \sum_{j=1}^{n/2-s'} x_{(v-1)n+j+s'} e^{i(n/2-s'-1+j)\theta} d\theta = 0
\end{aligned} \tag{2.9}$$

and

$$\begin{aligned}
& \int_{-\pi}^{\pi} \mathbf{p}_2(\theta)^t G_m(\theta) \overline{\mathbf{p}_1(\theta)} d\theta = \int_{-\pi}^{\pi} \sum_{u=1}^m \sum_{v=1}^m \check{\mathbf{p}}_{u2}(\theta) f_{u,v}(\theta) \overline{\check{\mathbf{p}}_{v1}(\theta)} d\theta \\
& = \int_{-\pi}^{\pi} \sum_{u=1}^m \sum_{v=1}^m f_{u,v}(\theta) e^{-n/2\theta} \sum_{j=1}^{n/2} x_{(u-1)n+n/2+j} e^{-i(j-1)\theta} \sum_{l=1}^{n/2} x_{(v-1)n+l} e^{i(l-1)\theta} d\theta \\
& = \sum_{u=1}^m \sum_{v=1}^m \int_{-\pi}^{\pi} f_{u,v}(\theta) e^{-i(2s'+1)\theta} \sum_{j=1}^{n/2-s'} x_{(u-1)n+n/2+s'+j} e^{-i(n/2-s'-1+j)\theta} \cdot \\
& \quad \cdot \sum_{l=1}^{n/2-s'} x_{(v-1)n+l} e^{ij\theta} d\theta = 0.
\end{aligned} \tag{2.10}$$

Since $F_m(\theta) - \gamma_1 G_m(\theta)$ is positive semi-definite, we have

$$\begin{aligned}
& \int_{-\pi}^{\pi} \mathbf{p}_1(\theta)^t [F_m - \gamma_1 G_m(\theta)](\theta) \overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta)^t [F_m(\theta) - \gamma_1 G_m(\theta)] \overline{\mathbf{p}_2(\theta)} d\theta \\
& \geq \int_{-\pi}^{\pi} \mathbf{p}_1(\theta)^t [F_m - \gamma_1 G_m(\theta)](\theta) \overline{\mathbf{p}_2(\theta)} + \mathbf{p}_2(\theta)^t [F_m(\theta) - \gamma_1 G_m(\theta)] \overline{\mathbf{p}_1(\theta)} d\theta.
\end{aligned} \tag{2.11}$$

By using Lemma 2.3, (2.9), (2.10) and (2.11), we get

$$\begin{aligned}
& \left| \frac{\mathbf{x}^t \tilde{T}_{n,m} \mathbf{x} - \mathbf{x}^t \tilde{B}_{n,m} \mathbf{x}}{\mathbf{x}^t \tilde{B}_{n,m} \mathbf{x}} \right| = \frac{\left| \int_{-\pi}^{\pi} (\mathbf{p}_1(\theta) F_m(\theta) \overline{\mathbf{p}_2(\theta)} + \mathbf{p}_2(\theta) F_m(\theta) \overline{\mathbf{p}_1(\theta)}) d\theta \right|}{\left| \int_{-\pi}^{\pi} (\mathbf{p}_1(\theta) F_m(\theta) \overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta) F_m(\theta) \overline{\mathbf{p}_2(\theta)}) d\theta \right|} \\
& = \frac{\left| \int_{-\pi}^{\pi} \mathbf{p}_1(\theta) [F_m(\theta) - \gamma_1 G_m(\theta)] \overline{\mathbf{p}_2(\theta)} + \mathbf{p}_2(\theta) [F_m(\theta) - \gamma_1 G_m(\theta)] \overline{\mathbf{p}_1(\theta)} d\theta \right|}{\left| \int_{-\pi}^{\pi} \mathbf{p}_1(\theta) F_m(\theta) \overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta) F_m(\theta) \overline{\mathbf{p}_2(\theta)} d\theta \right|} \\
& \leq \frac{\left| \int_{-\pi}^{\pi} \mathbf{p}_1(\theta) [F_m(\theta) - \gamma_1 G_m(\theta)] \overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta) [F_m(\theta) - \gamma_1 G_m(\theta)] \overline{\mathbf{p}_2(\theta)} d\theta \right|}{\left| \int_{-\pi}^{\pi} \mathbf{p}_1(\theta) F_m(\theta) \overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta) F_m(\theta) \overline{\mathbf{p}_2(\theta)} d\theta \right|} \\
& \leq 1 - \frac{\gamma_1}{\gamma_2} \quad \forall \mathbf{x} \in \Omega.
\end{aligned}$$

Therefore, we have

$$\alpha = \frac{\gamma_1}{\gamma_2} \leq \frac{\mathbf{x}^t \tilde{A}_{n,m} \mathbf{x}}{\mathbf{x}^t \tilde{B}_{n,m} \mathbf{x}} \leq 2 - \frac{\gamma_1}{\gamma_2} = \beta, \quad \forall \mathbf{x} \in \Omega.$$

It implies that there are at most $2ms'$ eigenvalues of $\tilde{B}_{n,m}^{-1} \tilde{A}_{n,m}$ outside the interval $[\alpha, \beta]$. \square

In [35], Serra explicitly constructed $G_m(\theta)$ by using eigen-decomposition of $F_m(\theta)$:

$$F_m(\theta) = Q(\theta)^* \Lambda(\theta) Q(\theta)$$

where $\Lambda(\theta)$ is a diagonal matrix containing the eigenvalues $\lambda_j(F_m(\theta))$ ($j = 1, \dots, m$) of $F_m(\theta)$. Suppose $\lambda_j(F_m(\theta))$ has a zero at θ_j of even order ν_j . Then $G_m(\theta)$ is constructed in the following way:

$$G_m(\theta) = \sum_{j=1}^m Q(\theta_j)^* \Gamma(\theta) Q(\theta_j),$$

where $\Gamma(\theta)$ is a diagonal matrix with

$$[\Gamma(\theta)]_{kk} = \begin{cases} (2 - 2 \cos(\theta))^{\nu_j/2}, & k = j \\ 1, & \text{otherwise.} \end{cases}$$

It is clear that each entry of $G_m(\theta)$ is a polynomial. The largest degree of the polynomials in $G_m(\theta)$ depends on the orders of the zeros of the eigenvalues of $F_m(\theta)$. It has been shown that $F_m(\theta)$ is spectrally equivalent to $G_m(\theta)$, see for instance [35].

Similarly, we one show the eigenvalues of $C_{n,m}^{-1} A_{n,m}$ are uniformly bounded except for a fixed number of outliers, where this fixed number depends on m .

THEOREM 2.5. *Let $F_m(\theta)$ be Hermitian positive definite. Suppose $F_m(\theta)$ is spectrally equivalent to $G_m(\theta) = [g_{u,v}]_{1 \leq u, v \leq m}$ where $g_{u,v}$ are trigonometric polynomials and s is the largest degree of the polynomials in $G_m(\theta)$. Then exist two positive numbers α and β ($\alpha < \beta$) independent of n such that for all $n > 2s'$ ($s' = \lceil s/2 \rceil$), at most ms' eigenvalues of $\tilde{C}_{n,m}^{-1} \tilde{A}_{n,m}$ (or $C_{n,m}^{-1} A_{n,m}$) are outside the interval $[\alpha, \beta]$.*

Proof. According to Theorems 2.2 and 2.3, when the eigenvalues of $B_{n,m}^{-1} A_{n,m}$ are equal to $1 - \lambda$, the eigenvalues of $C_{n,m}^{-1} A_{n,m}$ are given by $1 - \lambda^2$. Using Theorem 2.4, we can find two positive numbers $\alpha = (\gamma_1/\gamma_2)^2$ and $\beta = 1$ such that the result holds. \square

3. Recursive Computation of $B_{n,m}^{-1}$ and $C_{n,m}^{-1}$. In the previous section, we have shown that both $B_{n,m}$ and $C_{n,m}$ are good preconditioners for $A_{n,m}$. However, the inverses of $B_{n,m}$ and $C_{n,m}$ involves the inverse of $A_{n/2,m}$. The computational cost is still expensive. In this section, we present a recursive method to construct the preconditioners $B_{n,m}$ and $C_{n,m}$ efficiently.

We remark that the inverse of a Toeplitz matrix can be reconstructed by a low number of columns. Gohberg and Semencul [18] and Trench [39] showed that if the (1,1)th entry of the inverse of a Toeplitz matrix is nonzero, then the first and last columns of the inverse of the Toeplitz matrix are sufficient for this purpose. In [39] a recursion formula was given in [18]. A nice matrix representation of the inverse well-known as *Gohberg-Semencul formula* was presented. In [21], an inversion formula was exhibited which works for every nonsingular Toeplitz matrix and uses the solutions of two equations, (the so-called fundamental equations,) where the right-hand-side of

one of them is a shifted column of the Toeplitz matrix. Later Ben-Artzi and Shalom [2], Labahn and Shalom [26], Ng et al. [30] and Heinig [20] studied the representation when the (1,1)th entry of the inverse of a Toeplitz matrix is zero. In [31], we have used the matrix representation of the inverse of a Toeplitz matrix to construct effective preconditioners for Toeplitz matrices.

For block-Toeplitz matrices, Gohberg and Heinig [19] also extended the Gohberg-Semencul formula to handle this case. It was shown that if $A_{n,m}$ is nonsingular and the following equations are solvable:

$$A_{n,m}U^{(n)} = E^{(n)} \quad \text{and} \quad A_{n,m}V^{(n)} = F^{(n)} \quad (3.1)$$

with

$$U^{(n)} = \begin{pmatrix} U_1^{(n)} \\ U_2^{(n)} \\ \vdots \\ U_n^{(n)} \end{pmatrix}, \quad V^{(n)} = \begin{pmatrix} V_1^{(n)} \\ V_2^{(n)} \\ \vdots \\ V_n^{(n)} \end{pmatrix}, \quad E^{(n)} = \begin{pmatrix} I_m \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad F^{(n)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I_m \end{pmatrix}.$$

Here $U_j^{(n)}$ and $V_j^{(n)}$ are m -by- m matrices and I_m is the identity matrix. Assume that $U_1^{(n)}$ and $V_n^{(n)}$ are nonsingular, the inverse of $A_{n,m}$ can be expressed as follows:

$$A_{n,m}^{-1} = \Psi_{n,m}W_{n,m}\Psi_{n,m}^* - \Phi_{n,m}Z_{n,m}\Phi_{n,m}^* \quad (3.2)$$

where $\Psi_{n,m}$ and $\Phi_{n,m}$ are mn -by- mn lower triangular block Toeplitz matrices given respectively by

$$\Psi_{n,m} = \begin{pmatrix} U_1^{(n)} & 0 & \cdots & 0 & 0 \\ U_2^{(n)} & U_1^{(n)} & 0 & & 0 \\ \vdots & U_2^{(n)} & U_1^{(n)} & 0 & \vdots \\ U_{n-1}^{(n)} & & \ddots & \ddots & 0 \\ U_n^{(n)} & U_{n-1}^{(n)} & \cdots & U_2^{(n)} & U_1^{(n)} \end{pmatrix}$$

and

$$\Phi_{n,m} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ V_1^{(n)} & 0 & 0 & & 0 \\ \vdots & V_1^{(n)} & 0 & 0 & \vdots \\ V_{n-2}^{(n)} & & \ddots & \ddots & 0 \\ V_{n-1}^{(n)} & V_{n-2}^{(n)} & \cdots & V_1^{(n)} & 0 \end{pmatrix}.$$

Moreover, $W_{n,m}$ and $Z_{n,m}$ are block-diagonal matrices:

$$W_{n,m} = \begin{pmatrix} (U_1^{(n)})^{-1} & & & 0 \\ & (U_1^{(n)})^{-1} & & \\ & & \ddots & \\ 0 & & & (U_1^{(n)})^{-1} \end{pmatrix}$$

$$Z_{n,m} = \begin{pmatrix} (V_n^{(n)})^{-1} & & & 0 \\ & (V_n^{(n)})^{-1} & & \\ & & \ddots & \\ 0 & & & (V_n^{(n)})^{-1} \end{pmatrix}.$$

For the preconditioners $B_{n,m}^{-1}$ and $C_{n,m}^{-1}$, the inverse of $A_{n/2,m}$ can be represented by the formula in (3.2). This formula can be obtained by solving the following two linear systems:

$$A_{n/2,m}U^{(n/2)} = E^{(n/2)} \quad \text{and} \quad A_{n/2,m}V^{(n/2)} = F^{(n/2)}.$$

These two systems can be solved efficiently by using the preconditioned conjugate gradient (PCG) method with $B_{n/2,m}$ or $C_{n/2,m}$ as preconditioners. The inverse of $A_{n/2,m}$ involved in the preconditioners $B_{n/2,m}$ and $C_{n/2,m}$ can be recursively generated by using (3.2) until the size of the linear system is sufficiently small. The procedures of recursive computation of $B_{n,m}$ and $C_{n,m}$ are described as follows:

Procedure **Input**($A_{n,m}, n$) **Output**($U^{(n)}, V^{(n)}$)

If $k \leq N$, **then**

solve two linear systems

$$A_{k,m}U^{(k)} = E^{(k)} \quad \text{and} \quad A_{k,m}V^{(k)} = F^{(k)}$$

exactly by direct methods;

else

compute $U^{(k/2)}$ and $V^{(k/2)}$ by calling the procedure with the input matrix $A_{k/2,m}$ and the integer $k/2$; **construct** $A_{k/2,m}^{-1}$ by using the output $U^{(k/2)}$ and $V^{(k/2)}$ via the formula in (3.2);

solve the two linear systems

$$A_{k,m}U^{(k)} = E^{(k)} \quad \text{and} \quad A_{k,m}V^{(k)} = F^{(k)}$$

by using the preconditioned conjugate gradient method with $B_{k,m}$ (or $C_{k,m}$) as the preconditioner.

We remark that if each block of the block-Toeplitz matrix $A_{n,m}$ is Hermitian, then we only need to solve one linear system $A_{n,m}U^{(n)} = E^{(n)}$ in order to represent the inverse of the block-Toeplitz matrix. In this case, the solution $V^{(n)}$ can be obtained by using $U^{(n)}$:

$$V^{(n)} = \begin{pmatrix} U_n^{(n)} \\ U_{n-1}^{(n)} \\ \vdots \\ U_1^{(n)} \end{pmatrix}.$$

3.1. Computational Cost. The main computational cost of the method comes from the matrix-vector multiplications $A_{n,m}X$, $B_{n,m}^{-1}X$ (or $C_{n,m}^{-1}X$) in each PCG iteration, where X is an mn -by- m vector. We note that $A_{n,m}X$ can be computed in $2m$ $2n$ -length FFTs by first embedding $A_{n,m}$ into a $2mn$ -by- $2mn$ block-circulant matrix and then carrying out the multiplication by using the decomposition of the

block-circulant matrix. Let $S_{n,m}$ be the circulant matrix with m -by- m matrix block element, one can find a permutation matrix $P_{n,m}$ such that

$$\bar{S}_{n,m} = (S_{i,j})_{m \times m} = P_{n,m}^t W_{n,m} P_{n,m}$$

is a circulant-block matrix, where $S_{i,j}$ be n -by- n circulant matrix. Let $S_{i,j}(:, 1)$ denotes the first column of the matrix $S_{i,j}$, it is known that $S_{i,j}$ can be diagonalized in $n \log n$ length FFT, i.e. $S_{i,j} = F^* \Lambda_{i,j} F$, where F and F^* are the Fourier transform matrix and the inverse Fourier transform matrix, and $\Lambda_{i,j} = \text{diag}(F \cdot S_{i,j}(:, 1))$. Thus we obtain

$$\begin{aligned} \bar{S}_{n,m} &= (I \otimes F^*) \begin{pmatrix} \Lambda_{11} & \Lambda_{11} & \cdots & \Lambda_{1m} \\ \Lambda_{21} & \Lambda_{22} & \cdots & \Lambda_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \Lambda_{m1} & \Lambda_{m2} & \cdots & \Lambda_{mm} \end{pmatrix} (I \otimes F) \\ &= (I \otimes F^*) P^t D P (I \otimes F) \end{aligned}$$

where $D = \text{diag}(D_1, D_2, \dots, D_n)$ is a block diagonal matrix, and $[D_k]_{ij} = [\Lambda_{ij}]_{kk}$, i.e., the (i, j) th entry of D_k is equal to the (k, k) th entry of Λ_{ij} . Therefore, the block-circulant matrix-vector multiplication can be obtained by

$$S_{n,m} X = P(I \otimes F^*) P^t D P (I \otimes F) P^t X.$$

We note that it requires $O(m^2 n \log n)$ operations to compute the block diagonal matrix D , the block diagonal matrix-vector multiplication requires $O(m^3 n)$ operations. Thus the overall multiplication requires $O(m^2 n \log n + m^3 n)$. For the preconditioner $B_{n,m}$ or $C_{n,m}$, we need to compute matrix-vector products $A_{n/2,m}^{-1} Y$, where Y is an $mn/2$ -by- m vector. According to (3.2), the inverse of a block-Toeplitz matrix can be written as the product of lower-triangular block-Toeplitz matrices. Therefore, the matrix-vector multiplication $A_{n/2,m}^{-1} Y$ can be computed by using FFTs by embedding such lower-triangular block-Toeplitz matrices into a block-circulant matrices. Such matrix-vector multiplication requires $O(m^2 n \log n + m^3 n)$ operations.

Now we estimate the total cost of recursive computation for solving two linear systems

$$A_{n,m} U^{(n)} = E^{(n)} \quad \text{and} \quad A_{n,m} V^{(n)} = F^{(n)}.$$

For simplicity, we assume $n = 2^\ell$. Suppose the number of iterations required for convergence in solving the two mn_j -by- mn_j linear systems

$$A_{n_j,m} U^{(n_j)} = E^{(n_j)} \quad \text{and} \quad A_{n_j,m} V^{(n_j)} = F^{(n_j)} \quad \text{where} \quad n_j = 2^{\nu-j+1}$$

are given by c_j for $j = 1, \dots, L$. We note that the smallest size of the system is equal to $N = n/2^{\nu-L}$. Therefore the total cost of the recursive computations of $B_{n,m}$ (or $C_{n,m}$) is about $\sum_{j=1}^L c_j f_j$ where f_j denotes the cost of each PCG iteration where the size of the system is n_j . Since the cost of a n_j -length FFT is roughly twice the cost of a $n_j m/2$ -length FFT, and the cost of each PCG iteration is $O(m^2 n_j \log n_j + m^3 n_j)$ operations. Hence the total cost of the recursive computation is roughly bounded by $O(\max_j \{c_j (m^2 n \log n + m^3 n)\})$.

Next, we compute the operations required for the circulant preconditioners. For the block-circulant matrix $S_{n,m}$, the solution of $S_{n,m} Z = B$ can be obtain by:

$$Z = S_{n,m}^{-1} B = P(I \otimes F^*) P^t D^{-1} P (I \otimes F) P^t B$$

n	B	C	M
128	9	4	10
256	9	5	10
512	10	5	10

TABLE 4.1

Number of iterations required for convergence.

In order to compute the inverse of D , $O(m^2n \log n + m^3n)$ operations are required. Moreover, the matrix-vector multiplication requires $O(m^3n)$ operations, thus $S_{n,m}^{-1}B$ can be computed in $O(m^2n \log n + m^3n)$ operations, which is the same complexity of our proposed method. In next section, we show that our proposed method is competitive to circulant preconditioners.

4. Numerical Results. In this section, we test our proposed method. The initial guess is the zero vector. The stopping criteria is

$$\|r_q\|_2 / \|r_0\|_2 \leq 1 \times 10^{-7},$$

where r_q is the residual vector at the q -th iteration of the PCG method. We use Matlab 6.1 to conduct the numerical tests. We remark that our preconditioners are constructed recursively. For instance, when we solve $A_{256,m}U^{(256)} = E^{(256)}$, the preconditioners are constructed by solving $A_{128,m}U^{(128)} = E^{(128)}$ and $A_{64,m}U^{(64)} = E^{(64)}$ using the preconditioned conjugate gradient method with the stopping criteria being τ , and using the direct solver for $A_{32,m}U^{(32)} = E^{(32)}$. In all the tests, the coarsest level is set to be $n = 32$.

In the first test, we consider the following example of generating function [35]:

$$\begin{pmatrix} 20 \sin^2(\theta/2) & |\theta|^{5/2} \\ |\theta|^{5/2} & 20 \sin^2(\theta/2) \end{pmatrix}.$$

Tables 4.1 shows the corresponding numbers of iterations required for the convergence using our proposed preconditioners B and C . As a comparison, the numbers of iterations by using the preconditioner M studied in [35] are also listed. Our proposed preconditioners are competitive to the preconditioner studied in [35]. We also remark that the construction of our proposed preconditioners does not require the knowledge of the underlying matrix generating function of block-Toeplitz matrices.

In the second test, we consider the following four examples.

Example 1:

$$F_3(\theta) = \begin{pmatrix} 2\theta^4 + 1 & |\theta|^3 & \theta^4 \\ |\theta|^3 & 3\theta^4 + 1 & |\theta| \\ \theta^4 & |\theta| & 2\theta^4 + 1 \end{pmatrix}$$

Example 2:

$$F_3(\theta) = \begin{pmatrix} \theta^4 + 1 & |\theta|^3 & |\theta| \\ |\theta|^3 & 2\theta^4 + 1 & \theta^2 \\ |\theta| & \theta^2 & 5|\theta| \end{pmatrix}$$

Example 3:

$$F_2(\theta) = \begin{pmatrix} 8\theta^2 & (\sin \theta)^4 \\ (\sin \theta)^4 & 8\theta^2 \end{pmatrix}$$

Example 4:

$$F_3(\theta) = \begin{pmatrix} |\theta| & (\sin \theta)^4 & 0 \\ (\sin \theta)^4 & \theta^2 & (\sin \theta)^8 \\ 0 & (\sin \theta)^8 & \theta^4 \end{pmatrix}$$

These generating functions are Hermitian matrix-valued functions. Also the generated block-Toeplitz matrices are positive definite. In Example 1, the generated block-Toeplitz matrices are well-conditioned. For Examples 2-5, the generating functions are singular at some points and therefore the corresponding block-Toeplitz matrices are ill-conditioned.

In Tables 4.2-4.5, we give the number of iterations required for convergence by using $B_{n,m}$ and $C_{n,m}$ as the preconditioners.

Here we set $\tau = 1 \times 10^{-7}$ in the recursive calculation of the preconditioner and the maximum number of iterations to be 1000. If the method does not converge within 1000 iterations, we specify “> 1000” in the tables. According to Tables 4.2-4.5, we see that the number of iterations for the non-preconditioned systems (the column “ I ”) increases when the size n increases. However, the number of iterations for the preconditioned systems (the columns “ B ” and “ C ”) decreases or almost keep constant when the size n increases in Examples 1-3. The performance of Schur complement preconditioners C is generally better than that of block diagonal preconditioners B . We also compare our preconditioners with block-circulant preconditioners, the columns “ S ” and “ T ” are the number of iterations required for the Strang and the T. Chan block-circulant preconditioners respectively. We note that the Strang block-circulant preconditioner may not be positive definite for the ill-conditioned matrix. Indeed there are several negative eigenvalues of the Strang block-circulant preconditioners in Examples 3 and 4. Even the Strang circulant preconditioned system converges, the solution may not be correct. We also see from Tables 4.4 and 4.5 that the T. Chan block-circulant preconditioner does not work.

In [11, 13], Chan et al. have constructed “best” circulant preconditioners by approximating the generating function with the convolution product that matches the zeros of the generating function. They showed that these circulant preconditioners are effective for ill-conditioned Toeplitz matrices. Here we also construct such “best” block-circulant preconditioners (the column “ K_i ” and i refers to the order of the kernel that we used) and test their performance. We note from Tables 4.2-4.5 that our proposed preconditioners perform quite well. For Example 4, the method with “best” block-circulant preconditioners do not converge within 1000 iterations.

Also we report the computational times required for convergence in Examples 1-4 in Tables 4.6-4.9 respectively. If the number of iterations is more than 1000, we specify “**” in the tables. We see that the computational times required by the block diagonal preconditioner and the Schur complement preconditioner are less than those of the block-circulant preconditioners especially when n is large. We also note from the tables that the performance of the Schur complement preconditioner is better than that of the block diagonal preconditioner.

To illustrate the fast convergence of the proposed method, in Table 4.10, we calculate the number of eigenvalues within the small interval for $n = 128$ in Examples 1-4. We find that the spectra of the preconditioned matrices $C_{n,m}^{-1}A_{n,m}$ and $B_{n,m}^{-1}A_{n,m}$ are more close to 1 than those of circulant preconditioners and no preconditioner.

Finally, we report that the numbers of iterations are about the same even when the stopping criteria τ of the preconditioned conjugate gradient method at each level

n	I	B	C	S	T	K_4	K_6	K_8
64	111	13	6	13	12	12	12	12
128	124	12	6	13	12	12	13	13
256	133	9	4	13	12	13	13	13
512	135	8	4	13	13	12	13	12
1024	138	5	2	13	13	13	13	13
2048	139	2	1	13	13	13	13	13
4096	140	2	1	13	13	13	13	13

TABLE 4.2

Number of iterations required for convergence in Example 1.

n	I	B	C	S	T	K_4	K_6	K_8
64	114	12	6	11	11	11	12	12
128	172	12	6	12	12	11	12	12
256	256	12	6	12	13	11	12	13
512	371	13	6	12	13	12	13	13
1024	526	13	6	12	14	12	13	14
2048	740	13	6	12	15	12	13	14
4096	> 1000	13	7	12	15	12	12	14

TABLE 4.3

Number of iterations required for convergence in Example 2.

n	I	B	C	S	T	K_4	K_6	K_8
64	165	9	4	10	23	13	14	16
128	354	9	5	10	30	12	13	15
256	742	10	5	11	40	12	12	13
512	> 1000	10	5	11	54	12	12	13
1024	> 1000	10	5	11	> 1000	12	12	13
2048	> 1000	10	5	11	> 1000	12	12	13
4096	> 1000	10	6	11	> 1000	13	13	15

TABLE 4.4

Number of iterations required for convergence in Example 3.

n	I	B	C	S	T	K_4	K_6	K_8
64	585	19	9	46	> 1000	22	> 1000	27
128	> 1000	20	10	> 1000	> 1000	> 1000	> 1000	> 1000
256	> 1000	24	11	> 1000	> 1000	> 1000	> 1000	> 1000
512	> 1000	30	13	> 1000	> 1000	> 1000	> 1000	> 1000
1024	> 1000	36	16	> 1000	> 1000	> 1000	> 1000	> 1000
2048	> 1000	39	25	> 1000	> 1000	> 1000	> 1000	> 1000
4096	> 1000	43	23	> 1000	> 1000	> 1000	> 1000	> 1000

TABLE 4.5

Number of iterations required for convergence in Example 4.

in the recursive calculation of the proposed preconditioners is 1×10^{-3} , 1×10^{-4} and 1×10^{-7} for the proposed preconditioners.

Next we consider two applications to block-Toeplitz systems arising from multi-channel least squares filtering and queueing networks.

Application I: Multichannel least squares filtering is a data processing method that

n	I	B	C	S	T	K_4	K_6	K_8
64	0.51	0.28	0.38	0.21	0.20	0.20	0.20	0.20
128	0.70	0.50	0.38	0.28	0.26	0.26	0.28	0.28
256	1.13	0.78	0.49	0.59	0.54	0.59	0.59	0.59
512	2.15	1.31	0.99	1.21	1.21	1.11	1.21	1.11
1024	4.72	2.19	0.99	3.05	3.05	3.05	3.05	3.05
2048	11.63	4.79	1.18	10.94	10.94	10.94	10.94	10.94
4096	29.04	5.23	3.21	27.37	27.37	27.37	27.37	27.37

TABLE 4.6

Computational times required for convergence in Example 1.

n	I	B	C	S	T	K_4	K_6	K_8
64	0.52	0.26	0.21	0.18	0.18	0.18	0.20	0.20
128	0.97	0.50	0.38	0.26	0.26	0.24	0.26	0.26
256	2.18	1.04	0.73	0.54	0.59	0.50	0.54	0.60
512	5.92	2.13	1.49	1.11	1.21	1.11	1.20	1.21
1024	17.98	5.70	2.94	2.82	3.29	2.82	3.05	3.29
2048	61.94	11.64	7.08	10.10	12.63	10.10	10.94	11.78
4096	**	34.02	22.50	25.27	31.58	25.27	25.27	29.48

TABLE 4.7

Computational times required for convergence in Example 2.

n	I	B	C	S	T	K_4	K_6	K_8
64	0.75	0.20	0.14	0.16	0.38	0.21	0.23	0.26
128	1.99	0.38	0.32	0.22	0.65	0.26	0.28	0.33
256	6.04	0.87	0.61	0.50	1.81	0.54	0.54	0.59
512	**	1.64	1.24	1.02	5.01	1.11	1.11	1.21
1024	**	4.39	2.45	2.58	**	2.82	2.82	3.05
2048	**	8.95	5.90	9.25	**	10.10	10.10	10.94
4096	**	26.17	19.29	23.16	**	27.37	27.37	31.58

TABLE 4.8

Computational times required for convergence in Example 3.

n	I	B	C	S	T	K_4	K_6	K_8
64	2.67	0.41	0.31	0.75	**	0.36	**	0.44
128	**	0.83	0.64	**	**	**	**	**
256	**	2.08	1.34	**	**	**	**	**
512	**	4.92	3.22	**	**	**	**	**
1024	**	15.79	7.83	**	**	**	**	**
2048	**	34.91	29.50	**	**	**	**	**
4096	**	112.53	73.93	**	**	**	**	**

TABLE 4.9

Computational times required for convergence in Example 4.

makes use of the signals from each of m channels. We represent this multichannel data by \mathbf{x}_t , where \mathbf{x}_t is a column vector whose elements are the signals from each channel. Since we are interested in digital processing methods, we suppose that the signals are sampled at discrete, equally spaced time points which are represented by the time index t . Without loss of generality, we require that t takes on successive integer values. If we let x_{it} represents the signal coming from the i th channel ($i = 1, 2, \dots, m$), the

	I	B	C	S	T	K_4	K_6	K_8
Example 1	2.60%	94.27%	98.44%	77.34%	30.99%	93.23%	87.76%	77.08%
Example 2	4.43%	94.79%	98.44%	84.90%	46.62%	94.27%	94.01%	85.16%
Example 3	0.00%	95.31%	99.22%	89.84%	56.64%	84.38%	81.64%	78.91%
Example 4	0.52%	93.23%	98.18%	81.77%	51.30%	79.95%	73.44%	70.05%

TABLE 4.10

The percentages of the number of eigenvalues within the interval of $[0.99, 1.01]$ for $n = 128$.

multichannel signal can be written as

$$\mathbf{x}_t = (x_{1,t}, x_{2,t}, \dots, x_{m,t})^T.$$

The filter is represented by the coefficients

$$S_1, S_2, \dots, S_n$$

where each coefficient S_k ($k = 1, 2, \dots, n$) is an n -by- m matrix. The multichannel signal \mathbf{x}_t received by the array system represents the input to the filter and the resulting output of the filter is a multichannel signal, which we denote by the column vector

$$\mathbf{y}_t = (y_{1,t}, x_{2,t}, \dots, x_{m,t})^T.$$

The relationship between input \mathbf{x}_t and output \mathbf{y}_t is given by the convolution formula

$$\mathbf{y}_t = S_1 \mathbf{x}_t + S_2 \mathbf{x}_{t-1} + \dots + S_n \mathbf{x}_{t-n+1}.$$

The determination of the filter coefficients is based on the concept of a desired output denoted by a column vector

$$\mathbf{z}_t = (z_{1,t}, z_{2,t}, \dots, z_{m,t})^T.$$

On each channel ($i = 1, 2, \dots, m$), there will be an error between the desired output \mathbf{z}_t and the actual output \mathbf{y}_t . The mean square value of this error is given by

$$\mathcal{E}[(\mathbf{z}_t - \mathbf{y}_t)^2].$$

The sum of the mean square errors for all the channels is

$$\sum_{i=1}^m \mathcal{E}[(\mathbf{z}_t - \mathbf{y}_t)^2].$$

The least squares determination of the filter coefficients requires that this sum is minimum. This minimization leads to a set of linear equations

$$\begin{pmatrix} R_0 & R_1 & \cdots & R_{n-1} \\ R_1 & R_0 & \cdots & R_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ R_{n-1} & R_{n-2} & \cdots & R_0 \end{pmatrix} \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{pmatrix} = \begin{pmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{pmatrix}, \quad (4.1)$$

where

$$R_j = \mathcal{E}[\mathbf{x}_t \mathbf{x}_{t-j}^T] \quad \text{and} \quad G_j = \mathcal{E}[\mathbf{z}_t \mathbf{x}_{t-j+1}^T].$$

\mathbf{X}_1	\mathbf{X}_{129}	\mathbf{X}_{257}	$\cdot \cdot \cdot \cdot$
\mathbf{X}_2	\mathbf{X}_{130}	\mathbf{X}_{258}	$\cdot \cdot \cdot \cdot$
\cdot	\cdot	\cdot	
\cdot	\cdot	\cdot	
\cdot	\cdot	\cdot	
\mathbf{X}_{128}	\mathbf{X}_{256}	\mathbf{X}_{384}	$\cdot \cdot \cdot \cdot$

FIG. 4.1. *Color image and data vectors.*

Here R_j is an m -by- m matrix and is the autocorrelation coefficients of the input signal \mathbf{x}_t and G_j is an n -by- m matrix and is the cross-correlation coefficients between the desired output \mathbf{z}_t and the input signal \mathbf{x}_t .

In the test, a 128-by-128 color image is used to generate the data points. We consider the pixel value of the color image to be \mathbf{x}_t ($t = 1, 2, \dots, 128^2$), see Figure 4.1. We remark that color can be regarded as a set of three images in their primary color components: red, green and blue. In the least squares filtering, there are three channels, i.e., $m = 3$. Our task is to generate the multichannel least squares filters such that the sum of the mean square errors for all the channels

$$\sum_{i=1}^m \mathcal{E}\{\mathbf{x}_{t+1} - [S_1\mathbf{x}_t + S_2\mathbf{x}_{t-1} + \dots + S_n\mathbf{x}_{t-n+1}]^2\}$$

is minimum. Such least squares filters have been commonly used in color image processing for coding and enhancement [27]. Table 4.11 shows the number of iterations required for convergence. Notice that the generating function of the block-Toeplitz matrices are unknown in this case. However, the construction of the proposed preconditioners only requires the entries of $A_{n,m}$ and does not require the explicit knowledge of the generating function $F_m(\theta)$ of $A_{n,m}$. We find that the generated block-Toeplitz matrices are very ill-conditioned. Therefore, the number of iterations required for convergence without preconditioning is very large, but the performance of the preconditioners $B_{n,m}$ and $C_{n,m}$ are very good.

We also generate more synthetic multichannel data sets to test the performance of our proposed method for larger m . Table 4.12 shows the number of iterations required for convergence. The results show that our proposed preconditioner performs quite well.

Application II: We next apply the preconditioning method to solve the steady-state probability distribution of a Markovian overflow queueing network with batch arrivals of customers. This is a non-symmetric problem. It is the 2-queue overflow network considered in [12, 15]. The network consists of two queues with exogenous Poisson batch arrivals and exponential servers. Whenever queue 2 is full, the arriving

n	m	I	B	C	S	T	K_4	K_6	K_8
16	3	166	18	9	168	22	36	39	50
32	3	725	26	13	>1000	21	32	36	43
64	3	725	26	13	>1000	15	29	31	37
128	3	>1000	60	30	>1000	42	>1000	>1000	>1000
256	3	>1000	85	40	> 1000	> 1000	>1000	>1000	>1000
512	3	> 1000	95	44	> 1000	> 1000	>1000	>1000	>1000
1024	3	> 1000	101	51	> 1000	> 1000	>1000	>1000	>1000

TABLE 4.11

Number of iterations required for convergence.

n	m	I	B	C	S	T	K_4	K_6	K_8
16	6	473	26	13	503	23	45	52	66
32	6	958	30	15	> 1000	28	42	43	56
64	6	> 1000	39	18	> 1000	28	39	41	50
128	6	> 1000	50	25	> 1000	44	> 1000	> 1000	> 1000
16	9	731	38	18	945	31	58	62	81
32	9	> 1000	42	21	> 1000	35	55	58	72
64	9	> 1000	53	25	> 1000	35	59	65	75
128	9	> 1000	70	35	> 1000	68	> 1000	> 1000	> 1000
16	12	989	44	22	> 1000	36	65	71	94
32	12	> 1000	50	25	> 1000	40	63	66	87
64	12	> 1000	64	31	> 1000	42	75	81	> 1000
128	12	> 1000	103	47	> 1000	> 1000	> 1000	> 1000	> 1000

TABLE 4.12

Number of iterations required for convergence.

customers will overflow to queue 1 if it is not yet full. Otherwise the customers will be blocked and lost, see Figure 4.2.

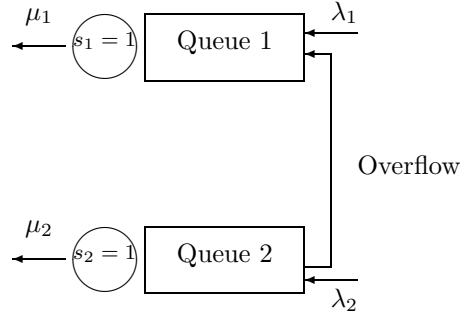


FIG. 4.2. The two-Queue overflow network.

For queue i ($i = 1, 2$), let λ_i be the exogenous input rate, μ_i the service rate, s_i the number of servers, and $n_i - s_i - 1$ the buffer sizes. For simplicity, we let the batch

size k of the arrivals follows the following distribution:

$$\text{Prob}(k = 1) = p_i \quad \text{and} \quad \text{Prob}(k = 2) = 1 - p_i \quad \text{where} \quad 0 < p_i < 1.$$

We assume that when the batch size is 2 and there is only one waiting space left, the queueing network still accepts one of the two customers. Then the generator matrix for the queueing system is given by

$$K = Q_1 \otimes I_{n_2} + I_{n_1} \otimes Q_2 + \text{diag}(0, \dots, 0, 1) \otimes R, \quad (4.2)$$

where

$$Q_i = \begin{pmatrix} \lambda_i & & & & & & & & 0 \\ -p_i \lambda_i & \lambda_i + \mu_i & & & & & & & \\ -(1-p_i) \lambda_i & \ddots & \ddots & & & & & & \\ & -(1-p_i) \lambda_i & -p_i \lambda_i & \lambda_i + \mu_i & & & & & \\ & & & \ddots & \ddots & & & & \\ & & & -(1-p_i) \lambda_i & -p_i \lambda_i & \lambda_i + \mu_i & & & \\ & & & & -(1-p_i) \lambda_i & -\lambda_i & \mu_i & & \\ 0 & & & & & & & & \end{pmatrix}, \quad (4.3)$$

$$R = \begin{pmatrix} \lambda_2 & & & & & 0 \\ -p_2 \lambda_2 & \lambda_2 & & & & \\ -(1-p_2) \lambda_2 & -p_2 \lambda_2 & \ddots & & & \\ & \ddots & \ddots & & & \\ 0 & & & -(1-p_2) \lambda_2 & -\lambda_2 & 0 \end{pmatrix} \quad (4.4)$$

and I_{n_i} is the identity matrix of size l_i . For the case of $p_i = 1$, the batch size is one, cosine-based preconditioners have been constructed to solve the queueing problem, see for instance [12, 15]. However, cosine-based preconditioners is not applicable when the input is a batch arrival Poisson process. Since the generator matrix is singular, we consider an equivalent linear system

$$(K + \mathbf{e}\mathbf{e}^t)\mathbf{p} = \mathbf{e} \quad (4.5)$$

Here $\mathbf{e} = (0, 0, \dots, 0, 1)^t$ and the $(K + \mathbf{e}\mathbf{e}^t)$ in (4.5) is irreducibly diagonal dominant and hence invertible. The steady-state distribution vector \mathbf{p} is obtained by normalizing \mathbf{p} .

We note that the matrix $(K + \mathbf{e}\mathbf{e}^t)$ can be expressed as the sum of a block-Toeplitz matrix

$$A = Q_1 \otimes I_{n_2} + I_{n_1} \otimes Q_2 + \text{diag}(\mu_1, 0, \dots, 0, \lambda_1) \otimes I_{n_2}$$

and a low-rank matrix

$$L = \begin{pmatrix} I_{n_2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} -\mu_1 I_{n_2} & 0 & \cdots & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I_{n_2} \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 & -\lambda_1 I_{n_2} + R + \text{diag}(0, \dots, 0, 1) \end{pmatrix}.$$

To solve (4.5), we can use the Sherman-Morrison-Woodbury formula for the matrix $A + L$ to solve for the vector \mathbf{p} . Since A is a block-Toeplitz matrix, our proposed method can be applied. Although A is not symmetric, the representation of the inverse of a nonsymmetric block-Toeplitz matrix is still defined (see the appendix). Table 4.13 shows the number of iterations required for convergence using the proposed method. We remark that the Strang preconditioner is singular in this application. We see that performance of our preconditioners is efficient.

p	$n = m$	I	B	C	T
0.4	4	16	10	5	10
0.4	8	30	12	6	13
0.4	16	54	14	7	16
0.4	32	87	16	8	20
0.8	4	14	10	5	10
0.8	8	31	13	6	14
0.8	16	57	15	7	18
0.8	32	103	17	8	23

TABLE 4.13

Number of Iterations required for the overflow queueing networks ($n_1 = n_2 = n$ and $p_1 = p_2 = p$).

5. Concluding Remarks. In this paper, we propose block diagonal and Schur complement preconditioners for block-Toeplitz matrices. We have proved that for some block-Toeplitz coefficient matrices, the spectra of the preconditioned matrices are uniformly bounded except for a fixed number of outliers, where the number of outliers depends on m . Therefore conjugate gradient method will converge very quickly when applied to solving the preconditioned systems, especially when m is small. Applications to block-Toeplitz systems arising from least squares filtering problems and queueing networks were discussed. The method can also be applied to solve other non-symmetric problems arise in other queueing systems [15].

Acknowledgements: The authors are very much indebted to the referees for their valuable comments and suggestions which greatly improved this paper.

REFERENCES

- [1] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, New York, 1994.
- [2] A. Ben-Artzi and T. Shalom, On inversion of Toeplitz and close to Toeplitz matrices, *Linear Algebra Appl.*, 75 (1986), 173–192.
- [3] F. Di Benedetto, *Analysis of preconditioning techniques for ill-conditioned Toeplitz matrices*, SIAM J. Sci. Comput., 16 (1995), pp. 682–697.
- [4] A. Ben-Artzi and T. Shalom, *On Inversion of Block Toeplitz Matrices*, Integral Equations and Operator Theory, 8 (1985), pp. 751–779.
- [5] F. Di Benedetto and S. Serra, *A Unifying Approach to Abstract Matrix Algebra Preconditioning*, Numer. Math., 82 (1999), pp. 57–90.
- [6] R. Chan, *Circulant Preconditioners for Hermitian Toeplitz Systems*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 542–550.
- [7] R. Chan, *Toeplitz Preconditioners for Toeplitz Systems with Nonnegative Generating Functions*, IMA J. Numer. Anal., 11 (1991), pp. 333–345.
- [8] R. Chan, Q. Chang and H. Sun, *Multigrid Method for Ill-Conditioned Symmetric Toeplitz Systems*, SIAM J. Sci. Comput., 19 (1998), pp. 516–529.
- [9] R. Chan and W. Ching, *Toeplitz-circulant Preconditioners for Toeplitz Systems and Their Applications to Queueing Network with Batch Arrivals*. SIAM J. Sci. Comp. ,17 (1996), pp. 427-482

- [10] R. Chan and M. Ng, *Conjugate Gradient Methods for Toeplitz Systems*, SIAM Review, 38 (1996), pp. 427–482.
- [11] R. Chan, M. Ng and A. Yip, *The Best Circulant Preconditioners for Hermitian Toeplitz Systems II: The Multiple-Zero Case*, Numer. Math., 92 (2002), pp. 17–40.
- [12] R. Chan, C. Wong and W. Ching *Optimal Trigonometric Preconditioners for Elliptic Problems and Queuing Problems*, SEA Bull. Math., 3 (1996), pp. 117–124.
- [13] R. Chan, A. Yip and M. Ng, *The Best Circulant Preconditioners for Hermitian Toeplitz Systems*, SIAM J. Numer. Anal., 38 (2001), pp. 876–896.
- [14] T. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 766–771.
- [15] W. Ching, *Iterative Methods for Queuing and Manufacturing Systems*, Springer Mathematics Monograph, Springer, London, 2001.
- [16] J. Durbin, *The Fitting of Time Series Models*, Rev. Int. Stat., 28 (1960), pp. 233–244.
- [17] G. Fiorentino and S. Serra, *Multigrid Methods for Symmetric Positive Definite Block Toeplitz Matrices with Nonnegative Generating Functions*, SIAM J. Sci. Comp., 17 (1996), pp. 1068–1081.
- [18] I. Gohberg and A. Semencul, *On the Inversion of Finite Toeplitz Matrices and Their Continuous Analogs*, Mat. Issled., 2 (1972), pp. 201–233.
- [19] I. Gohberg and G. Heinig, *Inversion of Finite-section Toeplitz Matrices Consisting of Elements of Non-commutative Algebra*, Rev. Roum. Math. Pures et Appl., 19 (1974), pp. 623–663.
- [20] G. Heinig, *On the Reconstruction of Toeplitz Matrix Inverses from Columns*, Linear Algebra Appl., 350 (2002), pp. 199–212.
- [21] G. Heinig and L. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Birkhäuser Verlag, Basel, 1984.
- [22] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, London, 1991.
- [23] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
- [24] T. Huckle, *Circulant and Skew Circulant Matrices for Solving Toeplitz Matrix Problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 767–777.
- [25] X. Jin, *Band-Toeplitz Preconditioners for Block Toeplitz Systems*, J. Comput. Appl. Math., 70 (1996), pp. 225–230.
- [26] G. Labahn and T. Shalom, *Inversion of Toeplitz matrices with only two standard equations*, Linear Algebra Appl., 175 (1992), pp. 143–158.
- [27] J. Lim, *Two-Dimensional Signal and Image Processing*, Prentice Hall, 1990.
- [28] M. Miranda and P. Tilli, *Asymptotic Spectra of Hermitian Block Toeplitz Matrices and Preconditioning Results*, SIAM J. Matrix Anal. Appl. 21 (2000), pp. 867–881.
- [29] M. Ng, *Band Preconditioners for Block-Toeplitz-Toeplitz-Block Systems*, Linear Algebra Appl., 259 (1997), pp. 307–327.
- [30] M. Ng, K. Rost and Y. Wen, *On Inversion of Toeplitz Matrices*, Linear Algebra Appl., 348 (2002), pp. 145–151.
- [31] M. Ng, H. Sun and X. Jin, *Recursive-based PCG Methods for Toeplitz Systems with Nonnegative Generating Functions*, SIAM Sci. Comput., 24, pp. 1507–1529.
- [32] D. Potts and G. Steidl, *Preconditioners for Ill-Conditioned Toeplitz Systems Constructed from Positive Kernels*, SIAM J. Sci. Comput., 22, pp. 1741–1761.
- [33] M. Priestley, *Spectral Analysis and Time Series*, Academic Press, 1981.
- [34] S. Serra, *Preconditioning Strategies for Asymptotically Ill-conditioned Block Toeplitz Systems*, BIT, 34 (1994), pp. 579–594.
- [35] S. Serra, *Spectral and Computational Analysis of Block Toeplitz Matrices Having Nonnegative Definite Matrix-Valued Generating Functions*, BIT, 39 (1999), pp. 152–175.
- [36] S. Serra, *Asymptotic Results on the Spectral of Block Toeplitz Preconditioned Matrices*, SIAM J. Matrix Anla. Appl., 20 (1998), pp. 31–44.
- [37] G. Strang, *A Proposal for Toeplitz Matrix Calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.
- [38] H. Sun, X. Jin and Q. Chang, *Convergence of the Multigrid Method for Ill-Conditioned Block Toeplitz Systems*, BIT, 41 (2001), pp. 179–190.
- [39] W. Trench, *An Algorithm for the Inversion of Finite Toeplitz Matrices*, SIAM J. Appl. Math., 12 (1964), pp. 515–522.
- [40] E. Tyrtyshnikov, *Optimal and Superoptimal Circulant Preconditioners*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 459–473.

Appendix. Let

$$A_{m,n} = \begin{pmatrix} A_0 & A_{-1} & \cdots & A_{1-n} \\ A_1 & A_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_{-1} \\ A_{n-1} & \cdots & A_1 & A_0 \end{pmatrix}$$

be a non-Hermitian block-Toeplitz matrix. We consider

$$\hat{A}_{m,n} = \begin{pmatrix} A_0^* & A_{-1}^* & \cdots & A_{1-n}^* \\ A_1^* & A_0^* & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_{-1}^* \\ A_{n-1}^* & \cdots & A_1^* & A_0^* \end{pmatrix}$$

and

$$M^{(n)} = \begin{pmatrix} 0 \\ A_{1-n} \\ \vdots \\ A_{-2} \\ A_{-1} \end{pmatrix} \quad \text{and} \quad \hat{M}^n = \begin{pmatrix} 0 \\ A_{1-n}^* \\ \vdots \\ A_{-2}^* \\ A_{-1}^* \end{pmatrix}.$$

Suppose the following matrix equations

$$A_{n,m}U^{(n)} = E^{(n)} \quad A_{n,m}V^{(n)} = M^{(n)}$$

and

$$\hat{A}_{n,m}\hat{U}^{(n)} = E^{(n)} \quad \hat{A}_{n,m}\hat{V}^{(n)} = \hat{M}^{(n)}$$

have the solutions:

$$U^{(n)} = \begin{pmatrix} U_1^{(n)} \\ U_2^{(n)} \\ \vdots \\ U_n^{(n)} \end{pmatrix} \quad V^{(n)} = \begin{pmatrix} V_1^{(n)} \\ V_2^{(n)} \\ \vdots \\ V_n^{(n)} \end{pmatrix} \quad \hat{U}^{(n)} = \begin{pmatrix} \hat{U}_1^{(n)} \\ \hat{U}_2^{(n)} \\ \vdots \\ \hat{U}_n^{(n)} \end{pmatrix} \quad \hat{V}^{(n)} = \begin{pmatrix} \hat{V}_1^{(n)} \\ \hat{V}_2^{(n)} \\ \vdots \\ \hat{V}_n^{(n)} \end{pmatrix}.$$

Then we have

$$A_{n,m}^{-1} = \begin{pmatrix} V_1^{(n)} & & & \\ V_2^{(n)} & V_1^{(n)} & & \\ \vdots & & \ddots & \\ V_n^{(n)} & \cdots & V_2^{(n)} & V_1^{(n)} \end{pmatrix} \begin{pmatrix} 0 & (\hat{U}_n^{(n)})^* & \cdots & (\hat{U}_2^{(n)})^* \\ & & \ddots & \vdots \\ & & & 0 & (\hat{U}_n^{(n)})^* \\ & & & & 0 \end{pmatrix} - \\ \begin{pmatrix} U_1^{(n)} & & & \\ U_2^{(n)} & U_1^{(n)} & & \\ \vdots & & \ddots & \\ U_n^{(n)} & \cdots & U_2^{(n)} & U_1^{(n)} \end{pmatrix} \begin{pmatrix} -I & (\hat{V}_n^{(n)})^* & \cdots & (\hat{V}_2^{(n)})^* \\ & & \ddots & \vdots \\ & & & -I & (\hat{V}_n^{(n)})^* \\ & & & & -I \end{pmatrix}.$$