Numer Algor (2017) 75:477–490
DOI 10.1007/s11075-016-0205-9

ORIGINAL PAPER

Circulant preconditioners for discrete ill-posed Toeplitz systems

L. Dykes[1,2] · S. Noschese[3] · L. Reichel[1]

Received: 10 June 2016 / Accepted: 4 September 2016 / Published online: 13 September 2016

Abstract Circulant preconditioners are commonly used to accelerate the rate of convergence of iterative methods when solving linear systems of equations with a Toeplitz matrix. Block extensions that can be applied when the system has a block Toeplitz matrix with Toeplitz blocks also have been developed. This paper is concerned with preconditioning of linear systems of equations with a symmetric block Toeplitz matrix with symmetric Toeplitz blocks that stem from the discretization of a linear ill-posed problem. The right-hand side of the linear systems represents available data and is assumed to be contaminated by error. These kinds of linear systems arise, e.g., in image deblurring problems. It is important that the preconditioner does not affect the invariant subspace associated with the smallest eigenvalues of the block Toeplitz matrix to avoid severe propagation of the error in the right-hand side. A perturbation result indicates how the dimension of the subspace associated with

Dedicated to Ken Hayami on the occasion of his 60th birthday.

✉ L. Reichel
reichel@math.kent.edu

L. Dykes
ldykes@math.kent.edu

S. Noschese
noschese@mat.uniroma1.it

[1] Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

[2] University School, Hunting Valley, OH 44022, USA

[3] Dipartimento di Matematica “Guido Castelnuovo”, SAPIENZA Università di Roma, P.le A. Moro, 2, I-00185 Roma, Italy

the smallest eigenvalues should be chosen and allows the determination of a suitable preconditioner when an estimate of the error in the right-hand side is available. This estimate also is used to decide how many iterations to carry out by a minimum residual iterative method. Applications to image restoration are presented.

Keywords Ill-posed problem · Deconvolution · FFT · Image deblurring

Mathematics Subject classification (2010) 65F10 · 65F15 · 65F30

1 Introduction

Linear systems of equations with a matrix with a Toeplitz-type structure arise in many applications, such as in signal and image processing. Consider the computation of an approximate solution of the linear system of equations

$$T\boldsymbol{x} = \boldsymbol{b}, \quad T \in \mathbb{R}^{n_1n_2\times n_1n_2}, \quad \boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^{n_1n_2}, \tag{1.1}$$

where T is a symmetric block Toeplitz with Toeplitz block (BTTB) matrix, i.e., T is a symmetric block Toeplitz matrix with each block being an $n_1 \times n_1$ symmetric Toeplitz matrix. The eigenvalues of T are assumed to decay smoothly to zero in magnitude without a significant gap. In particular, T may be singular. Linear systems of (1.1) with a matrix of this kind arise, for example, from the discretization of linear ill-posed problems, such as Fredholm integral equations of the first kind in two space-dimensions with a displacement kernel.

The right-hand side $\boldsymbol{b}$ of (1.1) is assumed to be contaminated by an (unknown) error $\boldsymbol{e}$. We will refer to this error as "noise." It may stem from measurement or discretization errors. Let $\widehat{\boldsymbol{b}}$ denotes the (unknown) error-free vector associated with $\boldsymbol{b}$, i.e.,

$$\boldsymbol{b} = \widehat{\boldsymbol{b}} + \boldsymbol{e}. \tag{1.2}$$

The (unknown) linear system of equations with error-free right-hand side,

$$T\boldsymbol{x} = \widehat{\boldsymbol{b}}, \tag{1.3}$$

is assumed to be consistent; however, we do not require the available system (1.1) to be consistent.

We will assume that a fairly sharp bound for the norm of $\boldsymbol{e}$ is known, i.e.,

$$\|\boldsymbol{e}\| \leq \varepsilon. \tag{1.4}$$

Here and throughout this paper, $\|\cdot\|$ denotes the Euclidean vector norm or the spectral matrix norm. The bound (1.4) will help us determine a suitable number of iterations to carry out with a minimal residual iterative method and to construct a preconditioner for the solution of (1.1).

Let $T^\dagger$ denotes the Moore–Penrose pseudoinverse of T. We are interested in computing an approximation of the solution $\widehat{\boldsymbol{x}} = T^\dagger\widehat{\boldsymbol{b}}$ of minimal Euclidean norm of the unavailable error-free linear system (1.3). Note that the solution of (1.1),

$$\boldsymbol{x} = T^\dagger\boldsymbol{b} = T^\dagger(\widehat{\boldsymbol{b}} + \boldsymbol{e}) = \widehat{\boldsymbol{x}} + T^\dagger\boldsymbol{e},$$

typically is dominated by the propagated error $T^{\dagger}\boldsymbol{e}$ and, therefore, is useless. Therefore, all solution methods for (1.1) seek to determine a suitable approximate solution that is not severely contaminated by propagated error. The computed approximate solution is the exact solution of an appropriately chosen nearby problem, whose solution is less sensitive to the error $\boldsymbol{e}$ in $\boldsymbol{b}$ than the solution of (1.1). The replacement of the given problem (1.1) by a nearby problem is commonly referred to as regularization. Among the most popular regularization methods are Tikhonov regularization, which replaces (1.1) by a penalized least-squares problem, and truncated iteration, which is based on solving (1.1) by an iterative method and terminating the iterations suitably early, see, e.g., [7, 10, 13, 17] for discussions of these regularization methods. In this paper, we regularize by truncated iteration and by choosing a suitable preconditioner.

The evaluation of matrix–vector products with a BTTB matrix of order n_1n_2 can be carried out in only $\mathcal{O}(n_1n_2\log_2(n_1n_2))$ arithmetic floating point operations (flops) by using the fast Fourier transform (FFT); see, e.g., [1, 12]. This makes it attractive to solve (1.1) by an iterative method. We will use a preconditioner to increase the rate of convergence of the iterative method. BTTB matrices are commonly preconditioned by block circulant with circulant block (BCCB) matrices; see [1, 3, 12, 19, 20] for discussions, illustrations, and further references. The use of BCCB preconditioners for BTTB matrices is attractive due to the spectral properties of the preconditioned matrix and because a matrix–vector product with a preconditioned matrix of order n_1n_2 can be evaluated in only $\mathcal{O}(n_1n_2\log_2(n_1n_2))$ flops with the aid of the FFT. A MATLAB software package for fast matrix–vector product evaluation is provided by Redivo–Zaglia and Rodriguez [16].

When preconditioning a BTTB matrix T that stems from the discretization of a linear ill-posed problem, it is desirable that an invariant subspace of T associated with the eigenvalues of smallest magnitude is not affected much by preconditioning to avoid severe propagation of the error $\boldsymbol{e}$ in the right-hand side $\boldsymbol{b}$ of (1.1) into the computed iterates. This is due to the fact that the eigenvectors associated with these eigenvalues are highly oscillatory (have many sign changes) and model noise rather than the desired solution $\widehat{\boldsymbol{x}}$. Typically, we do not want these eigenvectors to be part of our computed approximation of $\widehat{\boldsymbol{x}}$. A nice introduction to BCCB preconditioners for the solution of discretized linear ill-posed problems with a Toeplitz-type matrix is presented by Hanke et al. [9].

The number of iterations has to be few enough to avoid severe propagation of the error $\boldsymbol{e}$ in $\boldsymbol{b}$ into the computed approximation of $\widehat{\boldsymbol{x}}$. The availability of the bound (1.4) and the consistency of (1.3) allow us to apply the discrepancy principle to determine a suitable number of iterations as well as to define the dimension of the invariant subspace that should not be affected much by the preconditioner. Roughly, the larger the error in $\boldsymbol{b}$, the larger should the dimension of the subspace that is not (or only minimally) affected by preconditioning be chosen.

Various other approaches to define BCCB preconditioners for the iterative solution of discretized linear ill-posed problems (1.1) with an error-contaminated right-hand side are described in the literature. For instance, Hanke and Nagy [8] apply the L-curve criterion to determine a subspace that should not be affected by the preconditioner. The L-curve criterion implicitly estimates the norm of the error in $\boldsymbol{b}$. This criterion is able to estimate the norm of the error fairly accurately in some situations, but it is not a reliable error estimator; see [13, 17] for discussions and illustrations. We therefore are interested in developing an approach for constructing BCCB preconditioners that is not based on the L-curve criterion. Hanke et al. [9] apply a discrete Picard condition to determine the dimension of the subspace that the BCCB preconditioner should leave invariant. This approach typically works quite well in an interactive computing environment that allows the determination of whether the discrete Picard condition holds by visual inspection; however, it is not straightforward to automatize. Di Benedetto et al. [4] propose the application of a so-called superoptimal BCCB preconditioner and do not explicitly choose the dimension of the subspace that should be unaffected by the preconditioner. This approach works well for some image restoration problems, but not for others; see the discussion in [4].

Preconditioning is most useful when the error $\boldsymbol{e}$ in $\boldsymbol{b}$ is of small relative norm, because then many steps of an iterative method may be required to determine an accurate approximation of $\widehat{\boldsymbol{x}}$. When the error $\boldsymbol{e}$ is large, only few steps can be carried out before the propagated error destroys the computed solution. Preconditioning then does not reduce the computational effort by much.

In this paper, we will use the bound (1.4) to determine both the BCCB preconditioner and the number of iterations to be carried out. A perturbation bound guides our choice of preconditioner. This is described in Section 2. A few computed examples are presented in Section 3, and concluding remarks can be found in Section 4.

2 Preconditioned iterative regularization

We discuss the construction of the preconditioner, the stopping criterion for the iterative method, and outline the minimal residual iterative methods used.

2.1 The BCCB preconditioner

Let for the moment $T \in \mathbb{R}^{n\times n}$ be a symmetric positive definite Toeplitz matrix and let $C \in \mathbb{R}^{n\times n}$ be the closest circulant matrix to T in the Frobenius norm. T. Chan [2] proposed the use of C as a preconditioner for T; see also [1, 12]. The eigenvalues of C are given by the discrete Fourier transform of the first column of C; their computation with the FFT requires only $\mathcal{O}(n\log_2(n))$ flops. Since we do not want the preconditioner to affect the invariant subspace of T associated with the smallest eigenvalues much, we set the $n-p$ eigenvalues of the smallest magnitude of C to unity for some suitable $0 \leq p \leq n$, analogously as in [8, 9]. We refer to this preconditioner as C_p. Section 2.3 describes how to determine p using the error bound (1.4).

The linear systems (1.1) of interest to us have a BTTB matrix T, i.e., T is the Kronecker product of two Toeplitz matrices

$$T = T_1 \otimes T_2, \quad T_1 \in \mathbb{R}^{n_1\times n_1}, \quad T_2 \in \mathbb{R}^{n_2\times n_2}. \tag{2.1}$$

We will determine a preconditioner that is the Kronecker product of two circulant matrices

$$C = C_{p_1} \otimes C_{p_2}, \quad C_{p_1} \in \mathbb{R}^{n_1 \times n_1}, \quad C_{p_2} \in \mathbb{R}^{n_2 \times n_2}, \tag{2.2}$$

where C_{p_j} is defined by first determining the closest circulant C_j to T_j in the Frobenius norm and then setting its $n_j - p_j$ eigenvalues of smallest magnitude to one for $j = 1, 2$. In this way, our preconditioner C does not affect the invariant subspace of T associated with the eigenvalues of smallest magnitude much. The eigenvectors of this subspace are highly oscillatory and primarily model noise and not the desired solution $\widehat{\boldsymbol{x}}$. This construction of C requires only $\mathcal{O}(n_1 n_2(\log_2(n_1) + \log_2(n_2))$ flops. We refer to the BCCB preconditioner so determined as C_{p_1,p_2}. The determination of this preconditioner is somewhat faster than of the BCCB preconditioner described in [9], because the latter requires that all its $n_1 n_2$ eigenvalues be formed.

2.2 Stopping criterion

Once the BCCB preconditioner C_{p_1,p_2} has been defined, we compute an approximate solution $\boldsymbol{y}_k$ of the preconditioned linear system of equations,

$$T C_{p_1,p_2}^{-1} \boldsymbol{y} = \boldsymbol{b}, \tag{2.3}$$

using one of the minimal residual iterative methods described in Section 2.5. As for the initial approximation of $\widehat{\boldsymbol{x}}$, define the BCCB matrix $\widetilde{C}_{p_1,p_2} = \widetilde{C}_{p_1} \otimes \widetilde{C}_{p_2}$, where $\widetilde{C}_{p_j}$ is obtained from C_j by setting the $n_j - p_j$ eigenvalues of smallest magnitude to zero, for $j = 1, 2$, and define

$$\boldsymbol{x}_0 = \widetilde{C}_{p_1,p_2}^{\dagger} \boldsymbol{b}, \quad \boldsymbol{r}_0 = \boldsymbol{b} - T\boldsymbol{x}_0. \tag{2.4}$$

The initial iterate then is chosen to be $\boldsymbol{y}_0 = C_{p_1,p_2} \boldsymbol{x}_0$.

Let $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ denote the computed iterates. The number of iterations to be carried out is determined with the aid of the discrepancy principle. This stopping criterion prescribes that the iterations be terminated as soon as an iterate $\boldsymbol{y}_k$ that satisfies

$$\|T C_{p_1,p_2}^{-1} \boldsymbol{y}_k - \boldsymbol{r}_0\| \leq \gamma \varepsilon \tag{2.5}$$

has been determined. Here, $\gamma \geq 1$ is a user-specified parameter that is independent of ε. Typically, γ is chosen close to unity when ε is known to be a fairly sharp upper bound for $\|\boldsymbol{e}\|$; cf. (1.4). We obtain the approximation

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + C_{p_1,p_2}^{-1} \boldsymbol{y}_k$$

of the desired vector $\widehat{\boldsymbol{x}}$.

2.3 Construction of the preconditioner

Let for the moment $T \in \mathbb{R}^{n \times n}$ be a Toeplitz matrix and let $C \in \mathbb{R}^{n \times n}$ be the closest circulant in the Frobenius norm. Order the eigenvalues of C according to

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0.$$

Assume that a bound (1.4) is known. Let p be the number of eigenvalues of the largest magnitude of C that are not set to unity. We choose $p = \lfloor \frac{3}{4} q \rfloor$, where q is the solution of the minimization problem

$$\min_{1 \le q < n} \frac{1}{|\lambda_q|} \left(\frac{|\lambda_{q+1}|}{|\lambda_1|} + \eta \right). \tag{2.6}$$

Here, $\lfloor \alpha \rfloor$ denotes the largest integer smaller than or equal to $\alpha \ge 0$ and $\eta = \varepsilon / \|\boldsymbol{b}\|$. This choice of p is suggested by the following perturbation result.

Proposition 2.1 *Given a rank-q matrix $A \in \mathbb{C}^{n \times n}$, $q \le n$, a vector $\boldsymbol{\beta} \neq 0$ in the range of A, and $\boldsymbol{\xi}$ such that $\boldsymbol{\xi} = A^{\dagger} \boldsymbol{\beta}$. Let ΔA, $\boldsymbol{\Delta \beta}$, and $\boldsymbol{\Delta \xi}$ satisfy*

$$(A + \Delta A)(\boldsymbol{\xi} + \boldsymbol{\Delta \xi}) = \boldsymbol{\beta} + \boldsymbol{\Delta \beta}.$$

Then

$$\frac{\|\boldsymbol{\Delta \xi}\|}{\|\boldsymbol{\xi}\|} \le \nu(A, A + \Delta A)\, \kappa(A) \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\boldsymbol{\Delta \beta}\|}{\|\boldsymbol{\beta}\|} \right), \tag{2.7}$$

where $\kappa(M) = \|M\| \|M^{\dagger}\|$ is the condition number of the matrix M, and $\nu(M, N)$ denotes the ratio between the smallest singular values of the matrices M and N.

Proof From $A\boldsymbol{\xi} = \boldsymbol{\beta}$ and $(A + \Delta A)\boldsymbol{\Delta \xi} = -\Delta A \boldsymbol{\xi} + \boldsymbol{\Delta \beta}$, one has

$$\boldsymbol{\Delta \xi} = (A + \Delta A)^{\dagger} \left(-\Delta A \boldsymbol{\xi} + \boldsymbol{\Delta \beta} \right).$$

Taking norms on both sides, one gets

$$\|\boldsymbol{\Delta \xi}\| \le \|(A + \Delta A)^{\dagger}\| \left(\|\Delta A\| \|\boldsymbol{\xi}\| + \|\boldsymbol{\Delta \beta}\| \right).$$

Hence,

$$\|\boldsymbol{\Delta \xi}\| \le \nu(A, A + \Delta A)\, \kappa(A) \left(\frac{\|\Delta A\| \|\boldsymbol{\xi}\|}{\|A\|} + \frac{\|\boldsymbol{\Delta \beta}\|}{\|A\|} \right).$$

Finally, dividing by $\|\boldsymbol{\xi}\|$ and exploiting the inequality $\|\boldsymbol{\beta}\| \le \|A\| \|\boldsymbol{\xi}\|$ yield (2.7). □

Let $\widetilde{C}_q$ be the circulant obtained by setting the $n - q$ eigenvalues of smallest magnitude of C to zero. In this context, one replaces A by $\widetilde{C}_q$ and $\boldsymbol{\beta}$ by $\boldsymbol{b}$ in Proposition 2.1. Now, letting $\Delta A = T - \widetilde{C}_q$ and $\boldsymbol{\Delta \beta} = -\boldsymbol{e}$, we obtain by (1.2) that $\boldsymbol{\Delta \xi} = \widehat{\boldsymbol{x}} - \widetilde{C}_q^{\dagger} \boldsymbol{b}$ satisfies the hypothesis of Proposition 2.1. Inequality (2.7) reads

$$\begin{aligned} \frac{\|\hat{\boldsymbol{x}} - \widetilde{C}_q^{\dagger} \boldsymbol{b}\|}{\|\boldsymbol{x}_0\|} &\le \nu(\widetilde{C}_q, T)\, \kappa(\widetilde{C}_q) \left(\frac{\|T - \widetilde{C}_q\|}{\|\widetilde{C}_q\|} + \frac{\|\boldsymbol{e}\|}{\|\boldsymbol{b}\|} \right) \\ &\le \nu(\widetilde{C}_q, T)\, \frac{|\lambda_1|}{|\lambda_q|} \left(\frac{\|T - \widetilde{C}_q\|}{|\lambda_1|} + \eta \right) \sim \frac{|\lambda_1|}{|\lambda_q|} \left(\frac{|\lambda_{q+1}|}{|\lambda_1|} + \eta \right). \end{aligned}$$

In the final estimate, we assume that the eigenvalues of T of smallest magnitude are close to $|\lambda_q|$, so that $\nu(\widetilde{C}_q, T) \sim 1$ and $\|T - \widetilde{C}_q\| \sim \|C - \widetilde{C}_q\|$. This discussion suggests the choice $p = q$, where q is determined by (2.6). However, since we do not know whether the eigenvalues of T of smallest magnitude are close to $|\lambda_q|$, we will choose $p = \lfloor 3q/4 \rfloor$ to secure that we do not "over-precondition" and thereby obtain a large propagated error in the computed approximation of $\widehat{\boldsymbol{x}}$. We remark that a "standard" circulant preconditioner $\widetilde{C}_n$ generally over-preconditions and gives a large propagated error in the computed solution.

We turn to BTTB matrices and first consider the matrix $T \otimes T$. Let C be the closest circulant to T in the Frobenius norm. Then $C \otimes C$ is the closest BCCB matrix to $T \otimes T$ in the Frobenius norm. Let q to be the solution of the minimization problem

$$\min_{1 \le q < n} \frac{1}{|\lambda_q|^2}\left(\frac{|\lambda_{q+1}|^2}{|\lambda_1|^2} + \eta\right), \tag{2.8}$$

where $\eta = \varepsilon/\|\boldsymbol{b}\|$. The choice $p = q$ is suggested by Proposition 2.1, where one replaces A by $\widetilde{C}_{p,p}$, the BCCB matrix obtained setting the $n - p$ eigenvalues of the smallest magnitude of each circulant matrix C to zero, and replaces $\boldsymbol{\beta}$ by $\boldsymbol{b}$, so that the vector $\boldsymbol{\xi}$ in Proposition 2.1 is given by $\boldsymbol{x}_0$ in (2.4) with $p_1 = p_2 = q$. In the computed examples, we will let $p_1 = p_2 = \lfloor 3q/4 \rfloor$ to avoid to over-precondition.

Finally, consider BTTB matrices of the form (2.1). To determine a BCCB preconditioner of the kind (2.2), we sort the eigenvalues of the circulant matrices C_j, $j = 1, 2$, according to

$$|\lambda_1^{(j)}| \ge |\lambda_2^{(j)}| \ge \cdots \ge |\lambda_{n_j}^{(j)}| \ge 0,$$

and let p_j be the number of eigenvalues of the largest magnitude of C_j that is not set to unity. Let the index pair $\{q_1, q_2\}$ solve the minimization problem

$$\min_{\substack{1 \le q_1 < n_1 \\ 1 \le q_2 < n_2}} \frac{1}{|\lambda_{q_1}^{(1)}||\lambda_{q_2}^{(2)}|}\left(\frac{|\lambda_{q_1+1}^{(1)}||\lambda_{q_2+1}^{(2)}|}{|\lambda_1^{(1)}||\lambda_1^{(2)}|} + \eta\right), \tag{2.9}$$

where $\eta = \varepsilon/\|\boldsymbol{b}\|$. Similarly as above, we let $p_j = \lfloor 3q_j/4 \rfloor$ for $j = 1, 2$.

2.4 Construction of the preconditioned system

The preconditioned matrix TC_{p_1,p_2}^{-1}, where T and C_{p_1,p_2} are given by (2.1) and (2.2), respectively, is constructed as follows:

Compute for $j = 1, 2$:

1. The closest circulant C_j to T_j in the Frobenius norm.
2. The FFT of the first column of the matrix C_j. This gives the eigenvalues $\lambda_1^{(j)}, \lambda_2^{(j)}, \ldots, \lambda_{n_j}^{(j)}$ of C_j. The eigenvectors are the columns of the Fourier matrix. Permute the columns of the Fourier matrix so that the eigenvalues are ordered according to decreasing magnitude,

$$|\lambda_1^{(j)}| \ge |\lambda_2^{(j)}| \ge \ldots \ge |\lambda_{n_j}^{(j)}|$$

and denote the permuted Fourier matrix by U_j. The columns of U_j generally become more oscillatory with increasing column number.

3. The truncation index p_j, $1 \le p_j \le n_j$, by using (2.9). This defines the diagonal matrices

$$\begin{aligned}\Lambda_{p_j} &= \mathrm{diag}[\lambda_1^{(j)}, \lambda_2^{(j)}, \dots, \lambda_{p_j}^{(j)}, 1, \dots, 1],\\ \widetilde{\Lambda}_{p_j} &= \mathrm{diag}[\lambda_1^{(j)}, \lambda_2^{(j)}, \dots, \lambda_{p_j}^{(j)}, 0, \dots, 0].\end{aligned}$$

Neither the matrices U_j nor Λ_{p_j} and $\widetilde{\Lambda}_{p_j}$ have to be explicitly formed.

We are now in a position to define the preconditioner and related matrices, but hasten to point out that these matrices do not have to be explicitly formed. Introduce

$$\begin{aligned}C_{p_j} &= U_j \Lambda_{p_j} U_j^*, & C_{p_j}^{-1} &= U_j \Lambda_{p_j}^{-1} U_j^*, & \widetilde{C}_{p_j}^{\dagger} &= U_j \widetilde{\Lambda}_{p_j}^{\dagger} U_j^*,\\ C_{p_1,p_2} &= C_{p_1} \otimes C_{p_2}, & C_{p_1,p_2}^{-1} &= C_{p_1}^{-1} \otimes C_{p_2}^{-1}, & \widetilde{C}_{p_1,p_2}^{\dagger} &= \widetilde{C}_{p_1}^{\dagger} \otimes \widetilde{C}_{p_2}^{\dagger},\end{aligned}$$

where the superscript * denotes transposition and complex conjugation. We compute the initial approximate solution $\boldsymbol{x}_0 = \widetilde{C}_{p_1,p_2}^{\dagger}\boldsymbol{b}$ in (2.4) without explicitly forming the matrix $\widetilde{C}_{p_1,p_2}^{\dagger}$. Indeed, the spectral factorization

$$C_{p_1,p_2} = C_{p_1} \otimes C_{p_2} = (U_1 \otimes U_2)(\Lambda_{p_1} \otimes \Lambda_{p_2})(U_1 \otimes U_2)^*$$

can be applied to evaluate $C_{p_1,p_2}\boldsymbol{y}$ for any $\boldsymbol{y} \in \mathbb{R}^{n_1 n_2}$ in $\mathcal{O}(n_1 n_2 \log_2(n_1 n_2))$ flops with the FFT, and the same holds for matrix–vector products with the matrices C_{p_1,p_2}^{-1} and $\widetilde{C}_{p_1,p_2}^{\dagger}$.

Krylov subspace methods for the iterative solution of (2.3) require matrix–vector product evaluations with the preconditioned matrix TC_{p_1,p_2}^{-1}. It is well known that these matrix–vector product evaluations can be carried out quickly with the aid of the FFT. We outline for completeness the evaluation of matrix–vector products with the matrix TC_p^{-1} in the simplified situation when $T \in \mathbb{R}^{n \times n}$ is a Toeplitz matrix and C_p is a circulant. We express T as a sum of a circulant C_0 and a skew-circulant C_π. This splitting and the spectral factorizations

$$C_0 = U\Lambda_0 U^*, \quad C_\pi = U_\pi \Lambda_\pi U_\pi^*,$$

where $U_\pi = \mathrm{diag}[1, e^{i\pi/n}, \dots, e^{(n-1)i\pi/n}]U$, yield

$$TC_p^{-1} = (C_0 + C_\pi)C_p^{-1} = (U\Lambda_0 U^* + U_\pi \Lambda_\pi U_\pi^*)U\Lambda_p^{-1}U^*.$$

The preconditioned linear system of equations $TC_p^{-1}\boldsymbol{x} = \boldsymbol{b}$ with $\boldsymbol{b} \in \mathbb{R}^n$ can be expressed in the form

$$(\Lambda_0 + U^* U_\pi \Lambda_\pi U_\pi^* U)\Lambda_p^{-1}\boldsymbol{y} = U^*\boldsymbol{b}, \qquad \boldsymbol{x} = U\boldsymbol{y},$$

which is used in the computations. Each iteration requires the evaluation of the FFT of four n vectors. The computation of these FFTs is the dominating computational work. We remark that the dominating computational effort to evaluate a matrix–vector product with the matrix T, which is required when solving the unpreconditioned system $T\boldsymbol{x} = \boldsymbol{b}$ by a Krylov subspace method, also is the computation of the FFT of four n vectors. Therefore, the number of iterations required by the iterative method is the proper measure of the computational effort both for preconditioned and unpreconditioned linear systems of equations. The situation is analogous when T is the tensor product of two Toeplitz matrices. We omit the details.

2.5 Range restricted GMRES and MINRES methods

The generalized minimal residual method (GMRES) is a popular iterative method for the solution of large linear systems of equations $A\boldsymbol{x} = \boldsymbol{b}$ with a square nonsingular nonsymmetric matrix A that arises from the discretization of a well-posed problem, such as a Dirichlet boundary value problem for an elliptic partial differential equation; see, e.g., Saad [18]. The kth iterate determined by GMRES solves the minimization problem

$$\min_{x \in \mathcal{K}_k(A, r_0)} \|A\boldsymbol{x} - \boldsymbol{r}_0\|,$$

where $\boldsymbol{x}_0$ is an initial approximate solution, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, and

$$\mathcal{K}_k(A, \boldsymbol{r}_0) = \operatorname{span}\{\boldsymbol{r}_0, A\boldsymbol{r}_0, \ldots, A^{k-1}\boldsymbol{r}_0\}$$

is a Krylov subspace.

It has been observed that a modification of GMRES, which we refer to as the range restricted GMRES method (RRGMRES), often yields a more accurate approximation of the desired solution $\widehat{\boldsymbol{x}}$ than (standard) GMRES when A stems from the discretization of a linear ill-posed problem and the right-hand side $\boldsymbol{b}$ is contaminated by error; see [6, 14]. The RRGMRES method determines iterates in shifted Krylov subspaces $\mathcal{K}_k(A, A^{\ell}\boldsymbol{r}_0)$, where $\ell \geq 1$ is a small integer. We propose that an RRGMRES method be used for the solution of the preconditioned problem

$$TC_{p_1,p_2}^{-1}\boldsymbol{y} = \boldsymbol{r}_0.$$

When the system to be solved has a symmetric matrix, a simplified variant of RRGMRES, referred to as the range restricted MINRES method, can be applied instead. The latter method only requires simultaneous storage of a few n vectors, the number of which is bounded independently of the number of iterations; see [5] for details.

3 Computed examples

The calculations of this section were carried out in MATLAB with machine epsilon about $2.2 \cdot 10^{-16}$. For all the examples, we chose $\gamma = 1$ in (2.5).

Example 1 This is an image deblurring test problem from the MATLAB package Regularization Tools [11]. The original image and the symmetric BTTB matrix are those defined by the MATLAB function blur.m. We choose the dimensions $n_1 = n_2 = 64$, half-bandwidth for each Toeplitz block specified by the parameter $band = 10$, and width of the Gaussian point spread specified by the parameter $sigma = \sqrt{5}$.

We add to the blurred image determined by blur.m a noise vector $\boldsymbol{e}$ with normally distributed random entries with mean zero. The vector is normalized to correspond to a specified noise level. The pixels of the noise- and blur-contaminated image are ordered column-wise and stored in the right-hand side vector $\boldsymbol{b} \in \mathbb{R}^{64^2}$. First consider 0.1 % noise. Then the parameter p for the proposed BCCB preconditioner has the value 14; it is defined using (2.8). The discrepancy principle prescribes $k = 18$

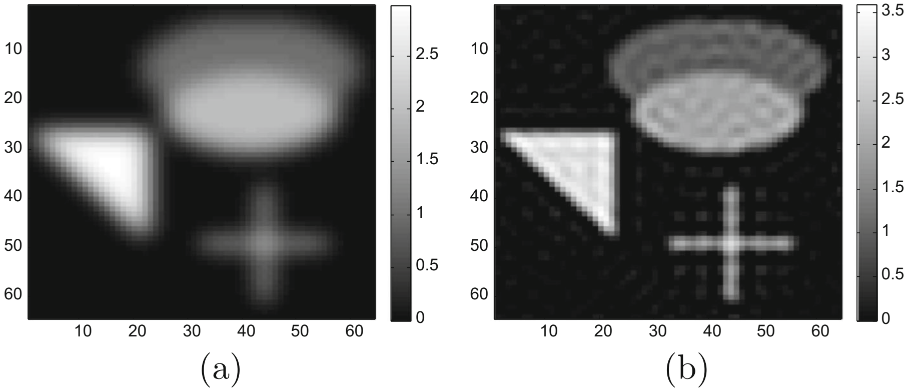

Fig. 1 Example 1. Noise level 0.1 %. Blurred and noisy image (**a**) and image restored with the proposed preconditioner (**b**)

iterations. This yields a restored image with relative error 0.3404. Figure 1 displays both the available blur- and noise-contaminated image and the computed restoration. When no preconditioner is used, the discrepancy principle terminates the iterations after $k = 33$ steps. The restoration so obtained has relative error 0.3361. It cannot be distinguished visually from the restoration determined by preconditioned iterations. We therefore do not display the former. We conclude that preconditioning reduces the number of iterations and therefore the computational effort by more than a half and gives a restoration of about the same quality as unpreconditioned iterations. Table 1 displays the p values used and the number of iterations required for 0.1, 0.05, and 0.01 % noise in $\boldsymbol{b}$. The noise- and blur-contaminated image together with the restoration determined by preconditioned iterations for the smallest noise level is displayed in Fig. 2.

Example 2 We use the same blur and relative noise as in the previous example, but now use the test image "Ken." For this image, $n_1 = n_2 = 136$. The BTTB matrix T was generated by the MATLAB function blur.m from [11] with the same parameter values as in the previous example. We add 0.1, 0.05, and 0.01 % white Gaussian noise to the blurred image to obtain a blur- and noise-contaminated image, which is

Table 1 Example 1: blur

% relative data error	p	steps k	$\|\boldsymbol{x}_k - \widehat{\boldsymbol{x}}\|/\|\widehat{\boldsymbol{x}}\|$
0.10	14	18	0.3404
0.10	–	33	0.3361
0.05	16	22	0.3308
0.05	–	45	0.3275
0.01	17	42	0.3094
0.01	–	89	0.3072

An en dash signifies that no preconditioner is used

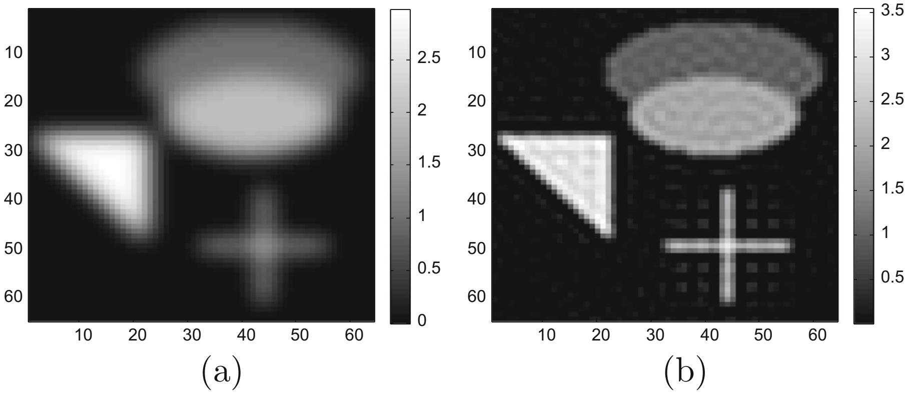

Fig. 2 Example 1. Noise level 0.01 %. Blurred and noisy image (**a**) and image restored with the proposed preconditioner (**b**)

stored in the right-hand side vector $\boldsymbol{b} \in \mathbb{R}^{136^2}$. Table 2 displays the number of iterations required with and without preconditioner to satisfy the discrepancy principle and the p-values that define the preconditioners for different noise levels. Figure 3 shows the contaminated and restored images for the noise level 0.1 % and Fig. 4 depicts the contaminated and restored images for the noise level 0.01 %. Similarly as for Example 1, preconditioned and unpreconditioned iterations give restorations of essentially the same quality. We therefore only show the restoration determined by preconditioned iterations.

Finally, consider the situation when the zero vector is chosen as an initial approximate solution for the preconditioned iterations instead of the vector (2.4). The parameter p that defines the preconditioner is given by (2.8) and the iterations are terminated by the discrepancy principle. Then for noise level 0.1 %, the discrepancy principle prescribes $k = 20$ iterations and gives a restoration with relative error 0.0904. The noise level 0.05 % requires $k = 21$ iterations and gives a restoration with relative error 0.0776, and the noise level 0.01 % demands $k = 28$ iterations and gives a restoration with relative error 0.0596. A comparison with Table 2 shows that the initialization (2.4) requires fewer iterations and gives restorations of higher quality than when using the initial vector $\boldsymbol{x}_0 = 0$.

Table 2 Example 2: “Ken”

% relative data error	p	steps k	$\|\boldsymbol{x}_k - \widehat{\boldsymbol{x}}\|/\|\widehat{\boldsymbol{x}}\|$
0.10	27	16	0.0764
0.10	–	21	0.0726
0.05	30	16	0.0691
0.05	–	28	0.0673
0.01	35	24	0.0571
0.01	–	54	0.0575

An en dash signifies that no preconditioner is used

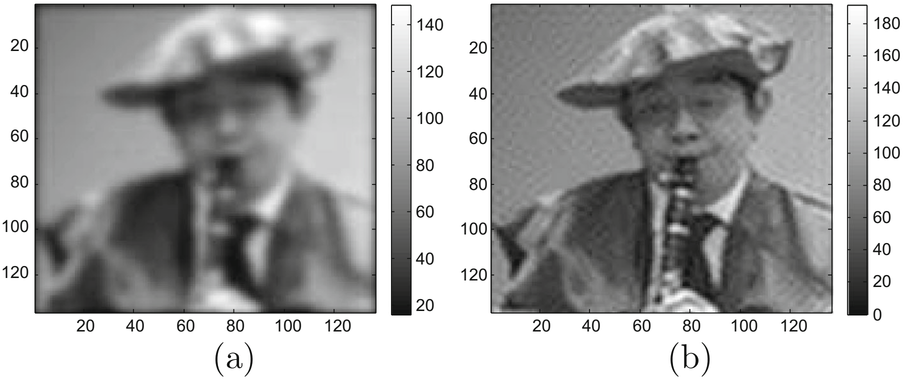

Fig. 3 Example 2. Noise level 0.1 %. Blurred and noisy image (**a**) and image restored with the proposed preconditioner (**b**)

Example 3 Our last example is the problem gravity from [11]. The linear system of equations (1.1) is obtained by discretizing an integral equation of the first kind with a space invariant kernel. This yields a Toeplitz matrix $T \in \mathbb{R}^{256\times 256}$ and right-hand side $\widehat{\boldsymbol{b}} \in \mathbb{R}^{256}$ to which we add an error vector $\boldsymbol{e} \in \mathbb{R}^{256}$ to obtain the right-hand side of (1.1); see (1.2). The error vector has normally distributed entries with mean zero and is scaled to correspond to the noise levels 0.1, 0.05, or 0.01 %. The noise level 0.1 % gives the parameter $p = 3$ for the circulant preconditioner and the discrepancy principle is satisfied after $k = 8$ iterations. We obtain the approximation $\boldsymbol{x}_8$ of the desired solution $\widehat{\boldsymbol{x}}$ with relative error 0.0144. Without a preconditioner, the same number of iterations is required to satisfy the discrepancy principle and the approximate solution obtained has a larger relative error, namely 0.0160. Table 3 summarizes the results for all noise levels considered. In this example, the preconditioners do not reduce the number of iterations required to satisfy the discrepancy principle, but improve the quality of the computed solutions.

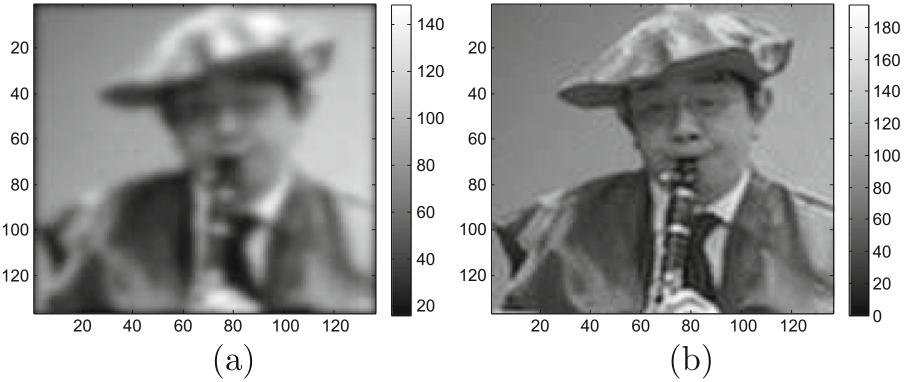

Fig. 4 Example 2. Noise level 0.01 %. Blurred and noisy image (**a**) and image restored with the proposed preconditioner (**b**)

Table 3 Example 3: gravity

% relative data error	p	steps k	$\lVert \boldsymbol{x}_k - \widehat{\boldsymbol{x}} \rVert / \lVert \widehat{\boldsymbol{x}} \rVert$
0.10	3	8	0.0144
0.10	–	8	0.0160
0.05	3	9	0.0105
0.05	–	9	0.0119
0.01	3	10	0.0077
0.01	–	10	0.0078

An en dash signifies that no preconditioner is used

In the computations reported in Table 3, we used the initial iterate (2.4). If instead the initial iterate $\boldsymbol{x}_0 = 0$ is used for the preconditioned iterations, then the discrepancy principle is for the noise level 0.1 % satisfied after $k = 9$ iterations and gives an approximate solution with relative error 0.0316. When the noise level is reduced to 0.05 %, the discrepancy principle prescribes that $k = 10$ iterations be carried out, resulting in an approximate solution with relative error 0.0222 and, finally, for noise level 0.01 %, $k = 11$ iterations are needed to satisfy the discrepancy principle and the computed approximate solution has relative error 0.0160. Thus, for all noise levels, the initial iterate $\boldsymbol{x}_0 = 0$ requires more iterations and gives approximate solutions of inferior quality than the initial iterate (2.4).

4 Conclusion and extension

This paper presents a novel method to determine BCCB preconditioners to be used for iterative solution of discretized linear ill-posed problem with a BTTB matrix. The computed examples show that the number of iterations is reduced by roughly a half when using the proposed preconditioner, whereas the quality of the computed solution is about the same as without preconditioning.

Also, we would like to mention that instead of using the circulant preconditioners described, one may use the generalized optimal circulant preconditioners described in [15] for the same computational cost. This may be attractive for certain Toeplitz and BTTB matrices.

References

1. Chan, R.H.-F., Jin, X.-Q.: An Introduction to Iterative Toeplitz Solvers, SIAM Philadelphia (2007)
2. Chan, T.: An optimal circulant preconditioner for Toeplitz systems. SIAM J. Sci. Statist. Comput. **9**, 766–771 (1988)
3. Chan, T., Olkin, J.A.: Circulant preconditioners for Toeplitz-block matrices. Numer. Algorithms **6**, 89–101 (1994)
4. Di Benedetto, F., Estatico, C., Serra Capizzano, S.: Superoptimal preconditioned conjugate gradient iteration for image deblurring. SIAM J. Sci. Comput. **26**, 1012–1035 (2005)
5. Dykes, L., Marcellán, F., Reichel, L.: The structure of iterative methods for symmetric linear discrete ill-posed problems. BIT Numer. Math. **54**, 129–145 (2014)

6. Dykes, L., Reichel, L.: A family of range restricted iterative methods for linear discrete ill-posed problems. Dolomites Research Notes on Approximation **6**, 27–36 (2013)
7. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems. Kluwer, Dordrecht (1996)
8. Hanke, M., Nagy, J.G.: Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques. Inverse Prob. **12**, 157–173 (1996)
9. Hanke, M., Nagy, J., Plemmons, R.: Ruttan Preconditioned iterative regularization for ill-posed problems. In: Reichel, L., Ruttan, A., Varga, R.S. (eds.) Numerical Linear Algebra, de Gruyter, pp. 141-163, Berlin (1993)
10. Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems, SIAM Philadelphia (1998)
11. Hansen, P.C.: Regularization tools version 4.0 for Matlab 7.3. Numer. Algorithms **46**, 189–194 (2007)
12. Ng, M.K.: Iterative Methods for Toeplitz Systems. Oxford University Press, Oxford (2004)
13. Kindermann, S.: Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems. Electron. Trans. Numer. Anal. **38**, 233–257 (2011)
14. Neuman, A., Reichel, L., Sadok, H.: Implementations of range restricted iterative methods for linear discrete ill-posed problems. Linear Algebra Appl. **436**, 3974–3990 (2012)
15. Noschese, S., Reichel, L.: The structured distance to normality of Toeplitz matrices with application to preconditioning. Numer Linear Algebra Appl. **18**, 429–447 (2011)
16. Redivo–Zaglia, M., Rodriguez, G.: smt: a Matlab toolbox for structured matrices. Numer. Algorithms **59**, 639–659 (2002)
17. Reichel, L., Rodriguez, G.: Old and new parameter choice rules for discrete ill-posed problems. Numer. Algorithms **63**, 65–87 (2013)
18. Saad, Y.: Iterative Methods for Sparse Linear Systems (2003)
19. van der Mee, C., Rodriguez, G., Seatzu, S.: Fast computation of two-level circulant preconditioners. Numer. Algorithms **41**, 275–295 (2006)
20. van der Mee, C., Rodriguez, G., Seatzu, S.: Fast superoptimal preconditioning of multiindex Toeplitz matrices. Linear Algebra Appl. **418**, 576–590 (2006)