Numerical Algorithms (2006) **41:** 275–295
DOI: 10.1007/s11075-005-9011-5

Fast computation of two-level circulant preconditioners

C. van der Mee, G. Rodriguez and S. Seatzu
Dipartimento di Matematica e Informatica, Università di Cagliari, viale Merello 92, 09123 Cagliari, Italy
E-mail: cornelis@bugs.unica.it; {rodriguez; seatzu}@unica.it

Received 11 February 2005; accepted 20 October 2005
Communicated by C. Brezinski

In this paper we present an algorithm for the construction of the superoptimal circulant preconditioner for a two-level Toeplitz linear system. The algorithm is fast, in the sense that it operates in FFT time. Numerical results are given to assess its performance when applied to the solution of two-level Toeplitz systems by the conjugate gradient method, compared with the Strang and optimal circulant preconditioners.

Keywords: Circulant preconditioning, multi index matrices.

AMS subject classification: 65F10, 15A69.

1. Introduction

A linear system

$$A\mathbf{x} = \mathbf{b},$$

with $\mathbf{x} = (x_0, \dots, x_{n-1})^T$ and $\mathbf{b} = (b_0, \dots, b_{n-1})^T$, is said to have Toeplitz structure if each entry of the matrix A depends only on the difference of its indices, that is if

$$A_{ij} = a_{i-j}, \quad i, j = 0, \dots, n-1.$$

Now, let $n = (n_1, n_2) \in \mathbb{Z}_+^2$ and let

$$\mathcal{I}_n = \left\{(i_1, i_2) \in \mathbb{Z}_+^2 : 0 \le i_1 \le n_1 - 1, 0 \le i_2 \le n_2 - 1\right\} \tag{1.1}$$

be a set of biindices. Similarly, a *two-level* Toeplitz system

$$A\mathbf{x} = \mathbf{b}, \tag{1.2}$$

with $\mathbf{x} = (x_i)_{i\in\mathcal{I}_n}$ and $\mathbf{b} = (b_i)_{i\in\mathcal{I}_n}$, is characterized by a matrix A such that

$$A_{ij} = a_{i-j}, \quad i, j \in \mathcal{I}_n.$$

In such a case the linear system can be expressed in the form

$$\sum_{j_1=0}^{n_1-1}\sum_{j_2=0}^{n_2-1} a_{(i_1-j_1,i_2-j_2)}x_{(j_1,j_2)} = b_{(i_1,i_2)},$$

with $i_1 = 0,\dots,n_1-1$ and $i_2 = 0,\dots,n_2-1$, and the Toeplitz matrix A can be conveniently stored in the array $G = (a_k)_{k\in\mathcal{J}_n}$, where

$$\mathcal{J}_n = \mathcal{I}_n - \mathcal{I}_n = \{(i_1,i_2)\in\mathbb{Z}^2 : -n_s+1 \le i_s \le n_s-1,\ s=1,2\}. \tag{1.3}$$

We will call $n = (n_1,n_2)$ the *dimension* of the linear system (1.2).

While there are many fast direct methods for the solution of one-level Toeplitz linear system, most of them have not yet been generalized to the multilevel case. On the other hand, the generalization of certain iterative methods, for example the conjugate gradient method, to multilevel linear systems is straightforward. Of course, the convergence of such iterative methods needs to be accelerated by employing suitable preconditioners.

Among the methods which have been proposed in the literature to generate circulant preconditioners, three are the most well-known, namely the Strang [21], optimal [9] and superoptimal [4, 23, 25] preconditioners. However, while the Strang and optimal preconditioners can be easily extended to the two-level case, this extension is more difficult for the superoptimal preconditioner. In particular, the method proposed in [25] does not easily generalize, as in the two-level case the product of two lower/upper triangular Toeplitz matrices is not a Toeplitz matrix and this property is used crucially in the algorithm proposed in [25]. In [11], a method to construct the superoptimal preconditioner is given, but only for Hermitian matrices.

In [16], introducing a new formalism, we obtain several properties concerning multiindex Toeplitz matrices which allow us to prove that the superoptimal circulant preconditioner of a multiindex Toeplitz matrix can be constructed with $O(N\log N)$ floating point operations, where N is the product of the components of the dimension $n = (n_1,\dots,n_d)$. However, in that paper no explicit algorithm is given.

Motivated by the apparent interest on this topic and by the lack of a general algorithm for the two-level case, in this paper we develop a method for generating the superoptimal preconditioner for a two-level Toeplitz matrix in $O(n_1n_2\log n_1n_2)$ floating point operations. The outline of the paper is as follows: in section 2 we recall the definition of some classical preconditioners and sum up briefly the two available procedures for the evaluation of the one-level superoptimal preconditioner. In section 3, we describe in detail our algorithm to compute in FFT time the superoptimal circulant preconditioner for a two-level Toeplitz linear system. Numerical results are then reported, in section 4, to assess the performance of the superoptimal preconditioner with respect to two other circulant preconditioners.

2. Circulant preconditioners

A circulant matrix [10] is a Toeplitz matrix $C = (c_{i-j})_{i,j=0}^{n-1}$ such that

$$c_{j-n} = c_j, \quad j = 1, \dots, n-1.$$

The idea of using a circulant matrix for preconditioning a Toeplitz linear system was first proposed by G. Strang [21] (see also [7]). In this case, the preconditioner is the circulant matrix $S = (s_{i-j})_{i,j=0}^{n-1}$ which agrees with a given Toeplitz matrix A on the widest possible band around the main diagonal. For example, if n is odd then

$$s_k = \begin{cases} a_k, & k = 0, \dots, \frac{n-1}{2} \\ a_{k-n}, & k = \frac{n+1}{2}, \dots, n-1. \end{cases}$$

Since then, the idea of extracting a preconditioner from an algebra of matrices whose elements can be diagonalized by a fast algorithm has been extended in many different directions. In particular, T. Chan in [9] proposed to consider the circulant matrix C which solves for a fixed matrix A, not necessarily structured, the optimization problem

$$\min \|C - A\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This preconditioner, which has been called *optimal*, if A is Toeplitz is given by

$$c_0 = a_0 \qquad \text{and} \qquad c_k = \frac{(n-k)a_k + k a_{k-n}}{n}, \quad k = 1, \dots, n-1. \tag{2.1}$$

Shortly after, different authors [4, 23, 25] introduced the so-called *superoptimal* preconditioner, defined to be the circulant matrix T which minimizes

$$\|I - T^{-1}A\|_F.$$

Also in this case, the matrix A need not be structured.

These preconditioners, when applied to a Toeplitz matrix and under mild assumptions, have been proved to have good properties in terms of clustering of the spectrum of the preconditioned matrix [3, 5, 7, 8] so that they guarantee superlinear convergence of the conjugate gradient iterative method [1, 19]. A review of the main results on these topics is contained in [6].

While the generation of the Strang preconditioner requires no computation and the optimal one can be computed with complexity $O(n)$ by a straightforward implementation of formulae (2.1), the construction of the superoptimal preconditioner, optimized with respect to its computational complexity, is not trivial. It turns out that this construction can be performed by $O(n \log_2 n)$ floating points operations (*flops*). In the following we review the two available procedures for its computation.

2.1. Tyrtyshnikov's construction

Let $D = (d_{i-j})_{i,j=0}^{n-1}$ be the circulant matrix which minimizes

$$\|I - D^{-1}A\|_F^2,$$

for a given matrix A. It has been proved [4, 25] that the first column $\mathbf{d}$ of D^{-1} is the solution of the linear system

$$\mathcal{C}_{(AA^*)}\mathbf{d} = \mathcal{C}_{(A^*)}\mathbf{e}_0, \tag{2.2}$$

where $\mathcal{C}$ is the linear operator which associates to a matrix A its optimal circulant preconditioner $\mathcal{C}_{(A)}$ and $\mathbf{e}_0 = (1, 0 \ldots, 0)^T$. Since this linear system is circulant it can be solved in FFT time, so the real issue is to compute the matrix $\mathcal{C}_{(AA^*)}$.

Let us now assume that $A = (a_{i-j})_{i,j=0}^{n-1}$ is Toeplitz, and decompose it as the sum of the Toeplitz matrices $L = (\ell_{i-j})_{i,j=0}^{n-1}$ and $U = (u_{i-j})_{i,j=0}^{n-1}$, lower and upper triangular, respectively, defined by

$$\ell_j = \begin{cases} \frac{a_0}{2}, & j = 0 \\ a_j & j > 0, \\ 0 & j < 0 \end{cases} \qquad u_j = \begin{cases} \frac{a_0}{2}, & j = 0 \\ a_j & j < 0\,. \\ 0 & j > 0 \end{cases}$$

From the linearity of $\mathcal{C}$ we get

$$\mathcal{C}_{(AA^*)} = \mathcal{C}_{(T)} + \mathcal{C}_{(LL^*)} + \mathcal{C}_{(UU^*)}$$

where $T = LU^* + UL^*$.

To obtain a fast algorithm for the computation of the first column $\mathbf{c}$ of $\mathcal{C}_{(AA^*)}$ a crucial property is that the product of two lower (or upper) triangular Toeplitz matrices is still Toeplitz. This guarantees that T is a Toeplitz matrix, so that it can be computed by simply multiplying L times the first column of U^* and taking into account that $UL^* = (LU^*)^*$.

Moreover, it is straightforward to verify that

$$\mathbf{c}^{(1)} := \mathcal{C}_{(LL^*)}\mathbf{e}_0 = \frac{1}{n}\left[SL\boldsymbol{\ell}^{(1)} + L^*\boldsymbol{\ell}^{(2)}\right]$$

$$\mathbf{c}^{(2)} := \mathcal{C}_{(UU^*)}\mathbf{e}_0 = \frac{1}{n}\left[SU^*\mathbf{u}^{(1)} + U\mathbf{u}^{(2)}\right]$$

where S is the *forward shift* matrix

$$S = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}$$

and

$$\begin{aligned} \ell_k^{(1)} &= (k+1)\,\overline{\ell_{n-k-1}}, & \ell_k^{(2)} &= (n-k)\,\ell_k, \\ u_k^{(1)} &= (k+1)\,u_{k-n+1}, & u_k^{(2)} &= (n-k)\,\overline{u_{-k}} \end{aligned}$$

for $k = 0, \ldots, n-1$. This shows that, letting $\mathbf{c}^{(0)} := \mathcal{C}_{(T)}\mathbf{e}_0$, the vector

$$\mathbf{c} := \mathcal{C}_{(AA^*)}\mathbf{e}_0 = \mathbf{c}^{(0)} + \mathbf{c}^{(1)} + \mathbf{c}^{(2)}$$

can be computed in $O(n \log_2 n)$ *flops*. More precisely, it requires 15 FFT's when A is a generic complex matrix and 10 if the matrix is Hermitian, including the solution of system (2.2).

2.2. *Chan/Tismenetsky's construction*

It is well-known that a circulant matrix can be diagonalized by means of the Fourier matrix. This allows us to factorize any circulant matrix $C = (c_{i-j})_{i,j=0}^{n-1}$ in the form

$$C = \mathcal{F}_\eta \Delta \mathcal{F}_\eta^*$$

where η is a primitive n-th root of unity (i.e., $\eta^j \neq 1$ for $j = 0, \ldots, n-1$ and $\eta^n = 1$), $\mathcal{F}_\eta = \frac{1}{\sqrt{n}}(\eta^{ij})_{i,j=0}^{n-1}$ is the normalized Fourier matrix,

$$\Delta = \operatorname{diag}\Big(\hat{C}(1), \hat{C}(\eta), \ldots, \hat{C}(\eta^{n-1})\Big)$$

and

$$\hat{C}(\zeta) = \sum_{k=0}^{n-1} c_k \zeta^{-k}$$

is the discrete Fourier transform of the first column of C.

Exploiting this factorization and the fact that the Frobenius norm is not modified by unitary transformations, we can write

$$\|I - D^{-1}A\|_F^2 = \|I - \Gamma B\|_F^2 \tag{2.3}$$

where the diagonal matrix $\Gamma = \operatorname{diag}(\gamma_0, \ldots, \gamma_{n-1})$ contains the eigenvalues of D^{-1} and $B = \mathcal{F}_\eta^* A \mathcal{F}_\eta$.

Equating to zero the gradient of (2.3) with respect to the real and imaginary parts of $\boldsymbol{\gamma} = (\gamma_p)_{p=0}^{n-1}$ we obtain

$$\gamma_p = \frac{\overline{b_{pp}}}{\sum_{j=0}^{n-1} |b_{pj}|^2} = \frac{(B^*)_{pp}}{(BB^*)_{pp}}, \quad p = 0, \ldots, n-1. \tag{2.4}$$

Chan/Tismenestky's approach for obtaining a fast algorithm for the computation of $\boldsymbol{\gamma}$ is based on the property that every Toeplitz matrix A can be expressed as the sum

$$A = C^{(0)} + C^{(1)}$$

of a circulant matrix $C^{(0)}$ and a skew-circulant matrix $C^{(1)}$, whose entries are given by

$$\begin{aligned} c_0^{(0)} &= a_0, \\ c_k^{(0)} &= c_{k-n}^{(0)} = \frac{a_k + a_{k-n}}{2}, \qquad k = 1, \ldots, n-1, \\ c_0^{(1)} &= 0, \\ c_k^{(1)} &= -c_{k-n}^{(1)} = \frac{a_k - a_{k-n}}{2}, \qquad k = 1, \ldots, n-1. \end{aligned}$$

The following factorizations of $C^{(0)}$ and $C^{(1)}$ hold

$$C^{(0)} = \mathcal{F}_\eta \Delta^{(0)} \mathcal{F}_\eta^*, \qquad C^{(1)} = D_\omega \mathcal{F}_\eta \Delta^{(1)} \mathcal{F}_\eta^* D_\omega^*,$$

where $D_\omega = \mathrm{diag}(1, \omega, \ldots, \omega^{n-1})$, $\omega^n = -1$ and the diagonal elements

$$\delta_k^{(0)} = \Delta_{kk}^{(0)} = \hat{C}^{(0)}(\eta^k), \quad \delta_k^{(1)} = \Delta_{kk}^{(1)} = \hat{C}^{(1)}(\omega\eta^k), \quad k = 0, \ldots, n-1,$$

are the discrete Fourier transforms of the first columns of $C^{(0)}$ and $D_\omega^* C^{(1)} D_\omega$, respectively. Employing these factorizations we get

$$A = \mathcal{F}_\eta \Big(\Delta^{(0)} E + E \Delta^{(1)} \Big) \mathcal{F}_\eta^* D_\omega^*,$$

where $E = \mathcal{F}_\eta^* D_\omega \mathcal{F}_\eta$ is a fixed unitary circulant matrix. This allows us to express the matrices appearing in (2.4) in the form

$$B = \mathcal{F}_\eta^* A \mathcal{F}_\eta = \Delta^{(0)} + E \Delta^{(1)} E^*$$

and

$$BB^* = \Delta^{(0)} {\Delta^{(0)}}^* + \Delta^{(0)} E {\Delta^{(1)}}^* E^* + E \Delta^{(1)} E^* {\Delta^{(0)}}^* + E \Delta^{(1)} {\Delta^{(1)}}^* E^*.$$

Further, as it is immediate to verify, for any diagonal matrix $\Delta = \mathrm{diag}(\delta_i)_{i=0}^{n-1}$ there holds

$$\Big\{ (E \Delta E^*)_{pp} \Big\}_{p=0}^{n-1} = M \boldsymbol{\delta},$$

where $M = (|E_{ij}|^2)_{i,j=0}^{n-1}$ is circulant and $\boldsymbol{\delta} = (\delta_0, \ldots, \delta_{n-1})^T$. Hence, denoting by $\mathbf{u} \circ \mathbf{v} = (u_0 v_0, \ldots, u_{n-1} v_{n-1})^T$ the Schur product of two vectors, we can write

$$\begin{aligned} \{(B)_{pp}\}_{p=0}^{n-1} &= \boldsymbol{\delta}^{(0)} + M \boldsymbol{\delta}^{(1)}, \\ \{(BB*)_{pp}\}_{p=0}^{n-1} &= \boldsymbol{\delta}^{(0)} \circ \overline{\boldsymbol{\delta}^{(0)}} + 2\mathrm{Re}\Big(\boldsymbol{\delta}^{(0)} \circ M \overline{\boldsymbol{\delta}^{(1)}} \Big) + M \Big(\boldsymbol{\delta}^{(1)} \circ \overline{\boldsymbol{\delta}^{(1)}} \Big). \end{aligned}$$

As a result, the computation of the vector $\boldsymbol{\gamma}$ involves only Schur products and multiplications by circulant matrices. By suitably optimizing the algorithm it is possible to compute the matrix D by 7 complex FFT's plus lower order operations. We remark that, in this as well as in the previous construction method, one further FFT can be saved, since to apply the preconditioner to an iterative method we just need its eigenvalues.

3. Two-level circulant preconditioners

In order to describe the techniques for constructing two-level circulant preconditioners, let us first introduce two-level circulant matrices and some of their properties, as well as generalize the notion of skew-circulant matrices. The superoptimal preconditioner of a two-level Hermitian Toeplitz matrix has recently been studied in [11], where a detailed analysis of its spectral properties is performed and it is numerically tested as a regularizing preconditioner for image deblurring.

Let $n = (n_1, n_2) \in \mathbb{Z}_+^2$ and let $\mathcal{I}_n$ be defined by (1.1). A two-level circulant matrix is a Toeplitz matrix $C = (c_{i-j})_{i,j\in\mathcal{I}_n}$ such that

$$c_{(k_1-n_1,k_2)} = c_{(k_1,k_2-n_2)} = c_{(k_1-n_1,k_2-n_2)} = c_{(k_1,k_2)}$$

for $k = (k_1, k_2) \in \mathcal{I}_n$, $k_1, k_2 \neq 0$, and

$$c_{(k_1-n_1,0)} = c_{(k_1,0)}, \quad c_{(0,k_2-n_2)} = c_{(0,k_2)},$$

for $k_1 = 1, \ldots, n_1 - 1$ and $k_2 = 1, \ldots, n_2 - 1$.

We say $\eta = (\eta_1, \eta_2) \in \mathbb{C}^2$ is a *primitive n-th root of unity* if $\eta_1^{n_1} = \eta_2^{n_2} = 1$ and

$$\eta_i^r \neq 1, \quad r = 1, \ldots, n_i - 1, \quad i = 1, 2.$$

In analogy with the one-index case, a two-level circulant matrix is diagonalized by the normalized two-level Fourier matrix $\mathcal{F}_\eta$ defined by

$$(\mathcal{F}_\eta)_{ij} = \frac{1}{\sqrt{n_1 n_2}} \eta_1^{i_1 j_1} \eta_2^{i_2 j_2}.$$

In fact, the following relation holds true

$$C\mathcal{F}_\eta = \mathcal{F}_\eta \operatorname{diag}\left(\widehat{C}(\eta_1^{k_1}\eta_2^{k_2})\right)_{k\in\mathcal{I}_n}$$

where

$$\widehat{C}(\zeta) = \widehat{C}(\zeta_1, \zeta_2) = \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} c_{(j_1,j_2)} \zeta_1^{-j_1} \zeta_2^{-j_2}$$

is the two-dimensional discrete Fourier transform of the first column $\mathbf{c} = (c_k)_{k\in\mathcal{I}_n}$ of C.

Now, let $\sigma = (\sigma_1, \sigma_2) \in \{0,1\}^2$. By a σ-circulant matrix we mean a Toeplitz matrix $C^{(\sigma)} = (c^{(\sigma)}_{i-j})_{i,j\in\mathcal{I}_n}$ such that $c^{(\sigma)}_k = 0$ for all $k \in \mathcal{I}_n$ with $k_r = 0$ whenever $\sigma_r = 1$ $(r = 1, 2)$ and

$$c^{(\sigma)}_{(k_1-\tau_1 n_1,\, k_2-\tau_2 n_2)} = (-1)^{\tau_1\sigma_1+\tau_2\sigma_2}\; c^{(\sigma)}_{(k_1,k_2)}$$

for the remaining values of k and for $\tau = (\tau_1, \tau_2) \in \{0,1\}^2$.

The matrix $C^{(0,0)}$ is a two-level circulant matrix, while for $\sigma \neq (0,0)$ we obtain generalizations of skew-circulant matrices. For example, if $\sigma = (0,1)$ we get $c^{(0,1)}_{(k_1,0)} = 0$ and, for $k_2 \neq 0$,

$$\begin{aligned} c^{(0,1)}_{(k_1,k_2-n_2)} &= -c^{(0,1)}_{(k_1,k_2)} \\ c^{(0,1)}_{(k_1-n_1,k_2)} &= c^{(0,1)}_{(k_1,k_2)} \\ c^{(0,1)}_{(k_1-n_1,k_2-n_2)} &= -c^{(0,1)}_{(k_1,k_2)}. \end{aligned}$$

For each $\sigma \in \{0,1\}^2$, we define the diagonal matrix

$$D^{(\sigma)}_\omega = \operatorname{diag}\Big(\omega_1^{\sigma_1 j_1}\omega_2^{\sigma_2 j_2}\Big)_{j\in\mathcal{I}_n},$$

where $\omega = (\omega_1, \omega_2) \in \mathbb{C}^2$ is such that $\omega_1^{n_1} = \omega_2^{n_2} = -1$. Then a σ-circulant matrix $C^{(\sigma)}$ can be diagonalized as follows

$$C^{(\sigma)} D^{(\sigma)}_\omega \mathcal{F}_\eta = D^{(\sigma)}_\omega \mathcal{F}_\eta \;\operatorname{diag}\;\Big(\hat{C}(\omega_1^{\sigma_1}\eta_1^{j_1}, \omega_2^{\sigma_2}\eta_2^{j_2})\Big)_{j\in\mathcal{I}_n},$$

where the diagonalizing matrix $D^{(\sigma)}_\omega \mathcal{F}_\eta$ is unitary.

As one can verify by direct computation, a two-level Toeplitz matrix can be decomposed into the following sum of σ-circulant matrices

$$A = C^{(0,0)} + C^{(0,1)} + C^{(1,0)} + C^{(1,1)}, \tag{3.1}$$

where

$$\begin{aligned} c^{(0,0)}_{(0,0)} &= a_{(0,0)} \\ c^{(0,0)}_{(0,k_2)} &= c^{(0,1)}_{(0,k_2)} = \frac{a_{(0,k_2)} - a_{(0,k_2-n_2)}}{2} \\ c^{(0,0)}_{(k_1,0)} &= c^{(1,0)}_{(k_1,0)} = \frac{a_{(k_1,0)} - a_{(k_1-n_1,0)}}{2} \\ c^{(0,1)}_{(0,0)} &= c^{(0,1)}_{(k_1,0)} = c^{(1,0)}_{(0,0)} = c^{(1,0)}_{(0,k_2)} = c^{(1,1)}_{(0,0)} = c^{(1,1)}_{(0,k_2)} = c^{(1,1)}_{(k_1,0)} = 0, \end{aligned}$$

for $k_1, k_2 \neq 0$, and

$$c^{(\sigma)}_k = \frac{1}{4} \sum_{\tau\in\{0,1\}^2} (-1)^{\tau_1\sigma_1+\tau_2\sigma_2}\; a_{(k_1-\tau_1 n_1,\, k_2-\tau_2 n_2)}$$

when $k_1, k_2 \neq 0$ and for each $\sigma \in \{0,1\}^2$.

For the σ-circulant matrices involved in decomposition (3.1) the following factorizations hold

$$\begin{aligned} C^{(0,0)} &= \mathcal{F}_\eta \Delta_0 \mathcal{F}_\eta^* \\ C^{(0,1)} &= (D_\omega^{(0,1)} \mathcal{F}_\eta) \Delta_1 (D_\omega^{(0,1)} \mathcal{F}_\eta)^* \\ C^{(1,0)} &= (D_\omega^{(1,0)} \mathcal{F}_\eta) \Delta_2 (D_\omega^{(1,0)} \mathcal{F}_\eta)^* \\ C^{(1,1)} &= (D_\omega^{(1,1)} \mathcal{F}_\eta) \Delta_3 (D_\omega^{(1,1)} \mathcal{F}_\eta)^*, \end{aligned} \tag{3.2}$$

where the diagonal matrices Δ_i, $i = 0, 1, 2, 3$, contain the eigenvalues of the corresponding σ-circulant matrices.

Extending the Strang and optimal preconditioners to two-level Toeplitz linear systems is relatively straightforward (see [9, 25]). We first need to introduce the notion of *admissible band*, that is a set $E \subset \mathcal{J}$ (see (1.3)) such that $i, j \in E$ and $((i_s - j_s)/n_s) \in \mathbb{Z}$ $(s = 1, 2)$ imply $i = j$. Then, fixing any admissible band E, the Strang preconditioner of a Toeplitz matrix A is the circulant matrix $S = (s_{i-j})_{i,\, j \in \mathcal{I}_n}$ which agrees with A on the band E, that is

$$s_i = \begin{cases} a_j, & i \in E \text{ and } ((i_s - j_s)/n_s) \in \mathbb{Z}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

The generalization of the admissible band adopted by Strang in [21] is the set

$$E = \left\{ (i_1, i_2) \in \mathbb{Z}^2 : \lfloor n_s/2 \rfloor - n_s + 1 \le i_s \le \lfloor n_s/2 \rfloor,\ s = 1, 2 \right\}.$$

As the two-level optimal preconditioner is concerned, the circulant matrix $C = (c_{i-j})_{i,\, j \in \mathcal{I}_n}$ which solves the problem

$$\min \|C - A\|_F,$$

for a given Toeplitz matrix A, is given by

$$c_{(i_1,i_2)} = \begin{cases} a_{(0,0)}, & i_1 = i_2 = 0, \\ \dfrac{(n_2 - i_2) a_{(0,i_2)} + i_2 a_{(0,i_2-n_2)}}{n_2}, & i_1 = 0, i_2 > 0, \\ \dfrac{(n_1 - i_1) a_{(i_1,0)} + i_1 a_{(i_1-n_1,0)}}{n_1}, & i_1 > 0, i_2 = 0, \\ \dfrac{\tilde{c}_{(i_1,i_2)}}{n_1 n_2}, & i_1, i_2 > 0, \end{cases} \tag{3.4}$$

where $0 \le i_1 \le n_1 - 1$, $0 \le i_2 \le n_2 - 1$ and

$$\begin{aligned} \tilde{c}_{(i_1,i_2)} = \quad & \big[(n_1 - i_1)(n_2 - i_2) a_{(i_1,i_2)} + (n_1 - i_1) i_2 a_{(i_1,i_2-n_2)} \\ & + i_1 (n_2 - i_2) a_{(i_1-n_1,i_2)} + i_1 i_2 a_{(i_1-n_1,i_2-n_2)} \big]. \end{aligned}$$

As in the one-index case, the construction of the first preconditioner requires no computation, while the second one can be obtained in $O(n_1 n_2)$ flops. We also note that both preconditioners are real whenever A is real and Hermitian whenever A is Hermitian.

The algorithm for the construction of the superoptimal circulant preconditioner is more involved and is described in the next section.

3.1. Construction of the superoptimal preconditioner

The natural approach to computing the superoptimal circulant preconditioner D of a two-level Toeplitz matrix $A = (a_{i-j})_{i,j\in\mathcal{I}_n}$ would be to generalize the one-level algorithms given in Sections 2.1 and 2.2.

Let us first take into consideration the first one. The first step consists of decomposing the matrix A into the sum of a lower triangular and an upper triangular Toeplitz matrix. In the two-level case, and in general in the multilevel case, the definition of a triangular matrix is not unique. One has to introduce a linear order $\preceq$ on $\mathbb{Z}^2$ which is compatible with addition, i.e., satisfying $i + l \preceq j + l$ whenever $i, j, l \in \mathbb{Z}^2$ and $i \preceq j$. Then, a matrix L (U) is lower (upper) triangular when $L_{ij} = 0$ for $i \prec j$ ($U_{ij} = 0$ for $i \succ j$).

In extending Tyrtyshnikov's algorithm we found a major difficulty in the fact that the product of two lower (upper) triangular multilevel Toeplitz matrices is not Toeplitz in general. As a simple counterexample consider the block Toeplitz matrix with Toeplitz blocks

$$L = \begin{bmatrix} L_0 & 0 \\ L_1 & L_0 \end{bmatrix}$$

where L_0 and L_1 are any Toeplitz matrices of dimension larger than 2 and L_0 is lower triangular. This matrix is what we get if we order the entries of a triangular two-level Toeplitz array with respect to the same linear order which defines its triangularity. Multiplying L by itself we do not obtain a two-level Toeplitz matrix (i.e., block Toeplitz with Toeplitz blocks) since the $(2,1)$-block $L_1 L_0 + L_0 L_1$ is not Toeplitz. For this reason, in the following we generalize the construction by Chan/Tismenetsky to two-level matrices.

Since equation (2.3) is formally identical in the two-level case, we can rewrite (2.4) as

$$\gamma_p = \frac{\overline{b_{pp}}}{\sum_{j\in\mathcal{I}_n} |b_{pj}|^2} = \frac{(B^*)_{pp}}{(BB^*)_{pp}}, \quad p \in \mathcal{I}_n, \tag{3.5}$$

where $B = \mathcal{F}_\eta^* A \mathcal{F}_\eta$ and $\Gamma = \operatorname{diag}(\gamma_p)_{p\in\mathcal{I}_n} = \mathcal{F}_\eta^* D^{-1} \mathcal{F}_\eta$ contains the eigenvalues of the inverse of the preconditioner D.

Using the factorizations (3.2) in the additive decomposition (3.1) of A we get

$$B = \Delta_0 + \widetilde{E}_2\Delta_1\widetilde{E}_2^* + \widetilde{E}_1\Delta_2\widetilde{E}_1^* + \widetilde{E}_0\Delta_3\widetilde{E}_0^*$$

where the matrices

$$\widetilde{E}_0 = E_1 \otimes E_2, \quad \widetilde{E}_1 = E_1 \otimes I_{n_2}, \quad \widetilde{E}_2 = I_{n_1} \otimes E_2$$

are unitary circulant and the elements of the matrices $E_s = \mathcal{F}_{\eta_s}^* D_{\omega_s}\mathcal{F}_{\eta_s}$ $(s = 1, 2)$ are given by

$$(E_s)_{ij} = \frac{2}{\eta_s(1 - \omega_s\eta_s^{j-i})}, \quad i, j = 0, \ldots, n_s - 1. \tag{3.6}$$

We recall that the tensor product of two matrices M and N is the biindex matrix defined by $(M \otimes N)_{ij} = M_{i_1,j_1}N_{i_2,j_2}$, $i, j \in \mathcal{I}_n$.

To obtain the quantities at the denominator of (3.5) we should compute the diagonal entries of the matrix

$$\begin{aligned} BB^* &= \Delta_0\Delta_0^* + \Delta_0\widetilde{E}_2\Delta_1^*\widetilde{E}_2^* + \Delta_0\widetilde{E}_1\Delta_2^*\widetilde{E}_1^* + \Delta_0\widetilde{E}_0\Delta_3^*\widetilde{E}_0^* \\ &+ (\Delta_0\widetilde{E}_2\Delta_1^*\widetilde{E}_2^*)^* + \widetilde{E}_2\Delta_1\Delta_1^*\widetilde{E}_2^* + \widetilde{E}_2\Delta_1(E_1 \otimes E_2^*)\Delta_2^*\widetilde{E}_1^* \\ &+ \widetilde{E}_2\Delta_1\widetilde{E}_1\Delta_3^*\widetilde{E}_0^* + (\Delta_0\widetilde{E}_1\Delta_2^*\widetilde{E}_1^*)^* + (\widetilde{E}_2\Delta_1(E_1 \otimes E_2^*)\Delta_2^*\widetilde{E}_1^*)^* \\ &+ \widetilde{E}_1\Delta_2\Delta_2^*\widetilde{E}_1^* + \widetilde{E}_1\Delta_2\widetilde{E}_2\Delta_3^*\widetilde{E}_0^* + (\Delta_0\widetilde{E}_0\Delta_3^*\widetilde{E}_0^*)^* \\ &+ (\widetilde{E}_2\Delta_1\widetilde{E}_1\Delta_3^*\widetilde{E}_0^*)^* + (\widetilde{E}_1\Delta_2\widetilde{E}_2\Delta_3^*\widetilde{E}_0^*)^* + \widetilde{E}_0\Delta_3\Delta_3^*\widetilde{E}_0^*. \end{aligned}$$

By examining the 16 terms in the previous expression one notices that they fall in five categories, each of which can be computed by means of FFT's and Schur products. In fact, letting

$$\boldsymbol{\delta}_k = (\delta_p^{(k)})_{p\in\mathcal{I}_n} = \Big((\Delta_k)_{pp}\Big)_{p\in\mathcal{I}_n}, \quad k = 0, 1, 2, 3,$$

it can be verified that

$$\begin{aligned} \{(\Delta_0\Delta_0^*)_{pp}\}_{p\in\mathcal{I}_n} &= \boldsymbol{\delta}_0 \circ \overline{\boldsymbol{\delta}_0} \\ \{(\Delta_0\widetilde{E}_i\Delta_k^*\widetilde{E}_i^*)_{pp}\}_{p\in\mathcal{I}_n} &= \boldsymbol{\delta}_0 \circ \widetilde{M}_i\overline{\boldsymbol{\delta}_k} \\ \{(\widetilde{E}_i\Delta_k\Delta_k^*\widetilde{E}_i^*)_{pp}\}_{p\in\mathcal{I}_n} &= \widetilde{M}_i(\boldsymbol{\delta}_k \circ \overline{\boldsymbol{\delta}_k}) \\ \{(\widetilde{E}_r\Delta_s\widetilde{E}_s\Delta_3^*\widetilde{E}_0^*)_{pp}\}_{p\in\mathcal{I}_n} &= \widetilde{M}_r(\boldsymbol{\delta}_s \circ \widetilde{M}_s\overline{\boldsymbol{\delta}_3}) \\ \{(\widetilde{E}_2\Delta_1(E_1 \otimes E_2^*)\Delta_2^*\widetilde{E}_1^*)_{pp}\}_{p\in\mathcal{I}_n} &= \widetilde{M}_2\boldsymbol{\delta}_1 \circ \widetilde{M}_1\overline{\boldsymbol{\delta}_2}, \end{aligned} \tag{3.7}$$

where $i = 0, 1, 2$, $k = 1, 2, 3$, $r, s = 1, 2$ and the matrices $\widetilde{M}_i$ with entries

$$(\widetilde{M}_i)_{pq} = |(\widetilde{E}_i)_{pq}|^2, \quad p, q \in \mathcal{I}_n,$$

are circulant. The matrices $\widetilde{M}_i$ can be evaluated by $O(n_1 n_2)$ operations using the expressions (3.6).

For the sake of clarity, we prove here the last of the equalities (3.7). First of all it should be noted that, for any diagonal matrix $D = \mathrm{diag}(d_{(k_1,k_2)})_{k\in\mathcal{I}_n}$, we have

$$(\widetilde{E}_1 D \widetilde{E}_2)_{pq} = (\widetilde{E}_0)_{pq} d_{(q_1,p_2)} \quad \text{and} \quad (\widetilde{E}_2 D \widetilde{E}_1)_{pq} = (\widetilde{E}_0)_{pq} d_{(p_1,q_2)},$$

for $p, q \in \mathcal{I}_n$. Then

$$\begin{aligned}(\widetilde{E}_2 \Delta_1 (E_1 \otimes E_2^*) \Delta_2^* \widetilde{E}_1^*)_{pp} &= [(\widetilde{E}_2 \Delta_1 \widetilde{E}_1)(\widetilde{E}_1 \Delta_2 \widetilde{E}_2)^*]_{pp} \\ &= \sum_{k\in\mathcal{I}_n} |(\widetilde{E}_0)_{pk}|^2 \delta^{(1)}_{(p_1,k_2)} \overline{\delta^{(2)}_{(k_1,p_2)}} \\ &= \sum_{k_1=0}^{n_1-1} |(E_1)_{p_1 k_1}|^2 \overline{\delta^{(2)}_{(k_1,p_2)}} \cdot \sum_{k_2=0}^{n_2-1} |(E_2)_{p_2 k_2}|^2 \delta^{(1)}_{(p_1,k_2)} \\ &= (\widetilde{M}_1 \overline{\boldsymbol{\delta}_2} \circ \widetilde{M}_2 \boldsymbol{\delta}_1)_p.\end{aligned}$$

The application of (3.7) leads us to the following expressions for the numerator and the denominator of (3.5):

$$\begin{aligned}\{(B)_{pp}\}_{p\in\mathcal{I}_n} &= \boldsymbol{\delta}_0 + \widetilde{M}_2\boldsymbol{\delta}_1 + \widetilde{M}_1\boldsymbol{\delta}_2 + \widetilde{M}_0\boldsymbol{\delta}_3 \\ \{(BB^*)_{pp}\}_{p\in\mathcal{I}_n} &= \boldsymbol{\delta}_0 \circ \overline{\boldsymbol{\delta}_0} + \widetilde{M}_2(\boldsymbol{\delta}_1 \circ \overline{\boldsymbol{\delta}_1}) + \widetilde{M}_1(\boldsymbol{\delta}_2 \circ \overline{\boldsymbol{\delta}_2}) + \widetilde{M}_0(\boldsymbol{\delta}_3 \circ \overline{\boldsymbol{\delta}_3}) \\ &\quad + 2\,\mathrm{Re}(\boldsymbol{\delta}_0 \circ \widetilde{M}_2\overline{\boldsymbol{\delta}_1}) + 2\,\mathrm{Re}(\boldsymbol{\delta}_0 \circ \widetilde{M}_1\overline{\boldsymbol{\delta}_2}) \\ &\quad + 2\,\mathrm{Re}(\boldsymbol{\delta}_0 \circ \widetilde{M}_0\overline{\boldsymbol{\delta}_3}) + 2\,\mathrm{Re}(\widetilde{M}_2\boldsymbol{\delta}_1 \circ \widetilde{M}_1\overline{\boldsymbol{\delta}_2}) \\ &\quad + 2\,\mathrm{Re}(\widetilde{M}_2(\boldsymbol{\delta}_1 \circ \widetilde{M}_1\overline{\boldsymbol{\delta}_3})) + 2\,\mathrm{Re}(\widetilde{M}_1(\boldsymbol{\delta}_2 \circ \widetilde{M}_2\overline{\boldsymbol{\delta}_3})).\end{aligned}$$

This shows that the two-level superoptimal circulant preconditioner can be computed in $O(n_1 n_2 \log_2 n_1 n_2)$ *flops*.

To speed up the computation we observe that since the matrices $\widetilde{M}_k$, $k = 0, 1, 2$, are tensor products, their eigenvalues can be evaluated by taking products of the eigenvalues of the matrices M_1 and M_2 defined by

$$(M_k)_{ij} = |(E_k)_{ij}|^2, \quad i, j = 0, \ldots, n_k - 1, \ k = 1, 2,$$

an operation which involves $O(n_1 n_2)$ *flops*.

A straightforward implementation of the previous formulae is contained in Algorithm 3.1. This algorithm takes as input the first columns $\mathbf{m}_i$ of the matrices M_1, M_2 and the first columns $\mathbf{c}_j^{(\omega)}$, $j = 0, 1, 2, 3$, of the two-level matrices $C^{(0,0)}$ and

$$(D_\omega^{(\sigma)})^* C^{(\sigma)} D_\omega^{(\sigma)}, \quad \sigma = (0,1), (1,0), (1,1),$$

coming from (3.2). All such vectors can be computed in $O(n_1 n_2)$ *flops*. The algorithm takes 25 two-dimensional FFT's and relies on the definition of *discrete Fourier* *transform* used in Matlab [15], namely if $\mathbf{x} = (x_{(k,\ell)})$ and $\mathbf{y} = (y_{(k,\ell)})$, with $k = 0, \dots, n_1 - 1$ and $\ell = 0, \dots, n_2 - 1$, then

$$(\texttt{fft2}(\mathbf{x}))_{(k,\ell)} = \sum_{r=0}^{n_1-1} \sum_{s=0}^{n_2-1} x_{(r,s)} \eta_1^{-kr} \eta_2^{-\ell s} = \sqrt{n_1 n_2} \cdot \mathcal{F}_\eta^* \mathbf{x}$$

and

$$(\texttt{ifft2}(\mathbf{y}))_{(k,\ell)} = \frac{1}{n_1 n_2} \sum_{r=0}^{n_1-1} \sum_{s=0}^{n_2-1} y_{(r,s)} \eta_1^{kr} \eta_2^{\ell s}, = \frac{1}{\sqrt{n_1 n_2}} \cdot \mathcal{F}_\eta \mathbf{y},$$

where $\eta_1 = \mathrm{e}^{2\pi i/n_1}$ and $\eta_2 = \mathrm{e}^{2\pi i/n_2}$. This means that if $C = \mathcal{F}_\eta \Delta \mathcal{F}_\eta^*$ is circulant, then

$$C\mathbf{x} = \texttt{ifft2}(\boldsymbol{\delta} \circ \texttt{fft2}(\mathbf{x})).$$

In the algorithm, the function $ones(m, n)$ returns an array of dimension $m \times n$ whose entries are all 1, $\texttt{conj}(\mathbf{x})$ gives the array $(\overline{x_{(k,\ell)}})$, the complex conjugate of $\mathbf{x}$, while $\mathrm{Re}(\mathbf{x})$ denotes the real part of $\mathbf{x}$.

input : n_1, n_2, $\mathbf{m}_i$ $(i = 1, 2)$, $\mathbf{c}_j^{(\omega)}$, $(j = 0, 1, 2, 3)$
for $j = 0, 1, 2, 3$
  $\boldsymbol{\delta}_j = \texttt{fft2}(\mathbf{c}_j^{(\omega)})$
$\widetilde{\mathbf{m}}_1 = \texttt{fft}(\mathbf{m}_1)$
$\widetilde{\mathbf{m}}_2 = \texttt{fft}(\mathbf{m}_2)$
$\widehat{\mathbf{m}}_1 = \widetilde{\mathbf{m}}_1 * \texttt{ones}(1, n_2)$
$\widehat{\mathbf{m}}_2 = \texttt{ones}(n_1, 1) * \widetilde{\mathbf{m}}_2^T$
$\widehat{\mathbf{m}}_0 = \widehat{\mathbf{m}}_1 \circ \widehat{\mathbf{m}}_2$
$\boldsymbol{\theta}_1 = \texttt{ifft2}(\widehat{\mathbf{m}}_2 \circ \texttt{fft2}(\boldsymbol{\delta}_1))$
$\boldsymbol{\theta}_2 = \texttt{ifft2}(\widehat{\mathbf{m}}_1 \circ \texttt{fft2}(\boldsymbol{\delta}_2))$
$\boldsymbol{\theta}_3 = \texttt{ifft2}(\widehat{\mathbf{m}}_0 \circ \texttt{fft2}(\boldsymbol{\delta}_3))$
$\boldsymbol{\gamma}_1 = \texttt{conj}(\boldsymbol{\delta}_0 + \boldsymbol{\theta}_1 + \boldsymbol{\theta}_2 + \boldsymbol{\theta}_3)$
$\mathbf{s}_1 = \texttt{ifft2}(\widehat{\mathbf{m}}_1 \circ \texttt{fft2}(\texttt{conj}(\boldsymbol{\delta}_3))$
$\mathbf{s}_2 = \texttt{ifft2}(\widehat{\mathbf{m}}_2 \circ \texttt{fft2}(\texttt{conj}(\boldsymbol{\delta}_3))$
$\widetilde{\boldsymbol{\gamma}}_2 = \mathrm{Re}(\boldsymbol{\delta}_0 \circ \texttt{conj}(\boldsymbol{\theta}_1)) + \mathrm{Re}(\boldsymbol{\delta}_0 \circ \texttt{conj}(\boldsymbol{\theta}_2))$
  $+ \mathrm{Re}(\boldsymbol{\delta}_0 \circ \texttt{conj}(\boldsymbol{\theta}_3)) + \mathrm{Re}(\boldsymbol{\theta}_1 \circ \texttt{conj}(\boldsymbol{\theta}_2))$
  $+ \mathrm{Re}(\texttt{ifft2}(\widehat{\mathbf{m}}_2 \circ \texttt{fft2}(\boldsymbol{\delta}_1 \circ \mathbf{s}_1)))$
  $+ \mathrm{Re}(\texttt{ifft2}(\widehat{\mathbf{m}}_1 \circ \texttt{fft2}(\boldsymbol{\delta}_2 \circ \mathbf{s}_2)))$
$\boldsymbol{\gamma}_2 = 2 * \widetilde{\boldsymbol{\gamma}}_2 + \boldsymbol{\delta}_0 \circ \texttt{conj}(\boldsymbol{\delta}_0)$
  $+ \texttt{ifft2}(\widehat{\mathbf{m}}_2 \circ \texttt{fft2}(\boldsymbol{\delta}_1 \circ \texttt{conj}(\boldsymbol{\delta}_1)))$
  $+ \texttt{ifft2}(\widehat{\mathbf{m}}_1 \circ \texttt{fft2}(\boldsymbol{\delta}_2 \circ \texttt{conj}(\boldsymbol{\delta}_2)))$
  $+ \texttt{ifft2}(\widehat{\mathbf{m}}_0 \circ \texttt{fft2}(\boldsymbol{\delta}_3 \circ \texttt{conj}(\boldsymbol{\delta}_3)))$
$\boldsymbol{\gamma} = \boldsymbol{\gamma}_1 \;./\; \boldsymbol{\gamma}_2$ (componentwise division)
$\mathbf{d} = \texttt{ifft2}(\boldsymbol{\gamma})$

Algorithm 3.1. Construction of the two-level superoptimal preconditioner.

$$
\begin{aligned}
&\textsf{input}: n_1, n_2, \mathbf{m}_i\ (i=1,2),\ \mathbf{c}_j^{(\omega)},\ (j=0,1,2,3)\\
&N = n_1 * n_2\\
&\textsf{for } j=0,1,2,3\\
&\qquad \boldsymbol{\delta}_j = \texttt{fft2}(\mathbf{c}_j^{(\omega)})\\
&\widetilde{\mathbf{m}}_1 = \texttt{ifft}(\mathbf{m}_1)/n_2\\
&\widetilde{\mathbf{m}}_2 = \texttt{ifft}(\mathbf{m}_2)/n_1\\
&\widehat{\mathbf{m}}_1 = \widetilde{\mathbf{m}}_1 * \texttt{ones}(1,n_2)\\
&\widehat{\mathbf{m}}_2 = \texttt{ones}(n_1,1) * \widetilde{\mathbf{m}}_2^T\\
&\widehat{\mathbf{m}}_0 = \widehat{\mathbf{m}}_1 \circ \widehat{\mathbf{m}}_2\\
&\boldsymbol{\theta}_1 = \texttt{fft2}(\widehat{\mathbf{m}}_2 \circ \mathbf{c}_1^{(\omega)})\\
&\boldsymbol{\theta}_2 = \texttt{fft2}(\widehat{\mathbf{m}}_1 \circ \mathbf{c}_2^{(\omega)})\\
&\boldsymbol{\theta}_3 = \boldsymbol{\theta}_1 + \boldsymbol{\theta}_2 + N * \texttt{fft2}(\widehat{\mathbf{m}}_0 \circ \mathbf{c}_3^{(\omega)})\\
&\boldsymbol{\gamma}_1 = \texttt{conj}(\boldsymbol{\delta}_0 + N * \boldsymbol{\theta}_3)\\
&\mathbf{s}_1 = N * \texttt{fft2}(\widehat{\mathbf{m}}_1 \circ \mathbf{c}_3^{(\omega)})\\
&\mathbf{s}_2 = N * \texttt{fft2}(\widehat{\mathbf{m}}_2 \circ \mathbf{c}_3^{(\omega)})\\
&\widetilde{\boldsymbol{\gamma}}_2 = \mathrm{Re}(\boldsymbol{\delta}_0 \circ \texttt{conj}(\boldsymbol{\theta}_3)) + N * \mathrm{Re}(\boldsymbol{\theta}_1 \circ \texttt{conj}(\boldsymbol{\theta}_2))\\
&\qquad + \mathrm{Re}(\texttt{fft2}(\widehat{\mathbf{m}}_2 \circ \texttt{ifft2}(\boldsymbol{\delta}_1 \circ \texttt{conj}(\mathbf{s}_1))))\\
&\qquad + \mathrm{Re}(\texttt{fft2}(\widehat{\mathbf{m}}_1 \circ \texttt{ifft2}(\boldsymbol{\delta}_2 \circ \texttt{conj}(\mathbf{s}_2))))\\
&\boldsymbol{\gamma}_2 = \boldsymbol{\delta}_0 \circ \texttt{conj}(\boldsymbol{\delta}_0)\\
&\qquad + N * \big(2 * \widetilde{\boldsymbol{\gamma}}_2 + \texttt{fft2}(\widehat{\mathbf{m}}_2 \circ \texttt{ifft2}(\boldsymbol{\delta}_1 \circ \texttt{conj}(\boldsymbol{\delta}_1)))\\
&\qquad + \texttt{fft2}(\widehat{\mathbf{m}}_1 \circ \texttt{ifft2}(\boldsymbol{\delta}_2 \circ \texttt{conj}(\boldsymbol{\delta}_2)))\\
&\qquad + N * \texttt{fft2}(\widehat{\mathbf{m}}_0 \circ \texttt{ifft2}(\boldsymbol{\delta}_3 \circ \texttt{conj}(\boldsymbol{\delta}_3)))\big)\\
&\boldsymbol{\gamma} = \boldsymbol{\gamma}_1\ ./\ \boldsymbol{\gamma}_2 \ \text{(componentwise division)}\\
&\mathbf{d} = \texttt{ifft2}(\boldsymbol{\gamma})
\end{aligned}
$$

Algorithm 3.2. Optimized construction of the superoptimal preconditioner.

The number of two-dimensional FFT's can be reduced to 20 by further optimizing the algorithm. The optimization is achieved by suitably economizing the computation by means of the identity

$$\texttt{ifft2}(\texttt{fft2}(\mathbf{x}) \circ \texttt{fft2}(\mathbf{y})) = n_1 n_2\ \texttt{fft2}(\texttt{ifft2}(\mathbf{x}) \circ \texttt{ifft2}(\mathbf{y})).$$

The resulting procedure is sketched in Algorithm 3.2.

4. Numerical results

To assess the effectiveness of circulant preconditioning of two-level Toeplitz linear systems we performed various numerical tests. Formulae (3.3), (3.4) and Algorithm 3.2 have been implemented in Matlab [15] and executed on a computer equipped with an AMD Athlon 64 3200+ processor and 1.5 GB RAM, running the GNU/Debian Linux 3.1 operative system [20].

In the numerical experiments we used two symmetric positive definite Toeplitz matrices. The first one is the *Gaussian matrix* $A = (a_{i-j})_{i,j\in\mathcal{I}_n}$ whose elements are

$$a_k = a_{(k_1,k_2)} = \sqrt{\frac{\det(\Sigma)}{2\pi}} \cdot \exp\left(-\frac{k^T\Sigma k}{2}\right), \quad k \in \mathcal{J}_n,$$

where

$$\Sigma = \begin{bmatrix} \sigma_1 & \theta \\ \theta & \sigma_2 \end{bmatrix}$$

is a positive definite parameter matrix [17]. If $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2)$ the Gaussian matrix is indeed a tensor product, since

$$a_{(k_1,k_2)} = a_{k_1} \cdot a_{k_2}, \quad a_{k_i} = \sqrt{\frac{\sigma_i}{2\pi}} \cdot \exp\left(-\frac{\sigma_i k_i^2}{2}\right),$$

while in the general case it is genuinely two-level. This matrix has a great importance in image restoration since it models atmospheric blurring [13].

The second example is a generalization of the *KMS* (Kac–Murdock–Szegő) matrix [14, 24], defined by

$$a_k = a_{(k_1,k_2)} = \rho^{|k_1|+|k_2|}, \quad \rho \in (0,1), \quad k \in \mathcal{J}_n,$$

which consists of the tensor product of two one-level KMS matrices.

The first test concerns computational complexity. Taking a two-level Toeplitz matrix of dimension $n = (n_1, n_2)$ with $n_1 = n_2 = 2^k$, $k = 5, \ldots, 11$, the superoptimal circulant preconditioner is constructed by Algorithm 3.2 and the execution time (in seconds) is recorded as the mean over 10 repeated experiments, in order to minimize the influence of system time. The results, shown in figure 1, give a numerical evidence of the $O(n_1 n_2 \log_2(n_1 n_2))$ complexity of the algorithm.

The effectiveness of the preconditioning procedures with respect to the condition number of a test matrix is investigated in table 1. Namely, we consider the Gaussian matrix of dimension $n = (5,5)$, letting $\Sigma = \sigma I$ with $\sigma = 0.2, 0.5, 1, 1.5, 2$. The matrix is first renumbered according to the lexicographic ordering on $\mathbb{Z}^2$, i.e.,

$$i \preceq j \iff i_1 < j_1 \text{ or } i_1 = j_1 \text{ and } i_2 \le j_2, \quad \forall i,j \in \mathcal{I}_2,$$

and then passed to the *cond* function of Matlab which returns the two-norm condition number. The same is done with the preconditioned matrices $S^{-1}A$, $C^{-1}A$ and $D^{-1}A$. Table 1 shows that there is a substantial reduction in the conditioning whenever $\mathrm{cond}(A)$ is small, while the circulant preconditioners are progressively less effective as the condition number increases.

The convergence rate of iterative methods in the solution of a linear system is strongly influenced by the location of the eigenvalues of the matrix, more than by the

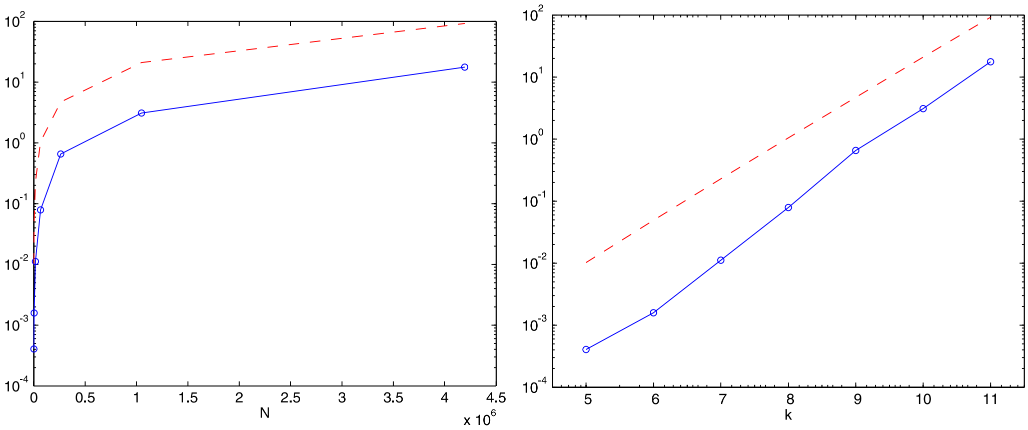

Figure 1. Execution time in seconds of Algorithm 3.2, with respect to the dimension $N = n_1 n_2 = 4^k$, $k = 5, \ldots, 11$, in semilogarithmic scale (*left*) and in loglog–scale (*right*); the *dashed line* is $10^{-6} N \log_2 N$.

mere condition number. In fact, it is known that if the spectrum of a Hermitian matrix A exhibits a proper cluster at 1 as the dimension n grows, that is if all its eigenvalues lie in a fixed neighbourhood of 1 except a finite number of *outliers* independent on n, then the conjugate gradient method applied to the linear system $A\mathbf{x} = \mathbf{b}$ converges superlinearly [1, 19] (see also [22]).

To illustrate the effect of preconditioning on the spectrum it is customary to draw in a graph the layout of the eigenvalues of the preconditioned matrices and of the original one. The top row in figure 2 shows a case in which the preconditioning procedures are successful (the dashed line marks $x = 1$). The test refers to the KMS matrix with $\rho = 0.5$ and $n = (5, 5)$ or $n = (20, 20)$. The eigenvalues are computed by the `eig` function of Matlab after reordering lexicographically the matrix. We can see that in both cases the 'clustering around one' effect takes place, so we expect a substantial reduction in the number of iterations of the conjugate gradient method. This outcome is rather obvious, since a two-level linear system of dimension $n = (n_1, n_2)$ whose matrix is a tensor product can always be decomposed by solving *in*

Table 1
Preconditioning of the Gaussian matrix A, with $n_1 = n_2 = 10$ and $\Sigma = \mathrm{diag}(\sigma, \sigma)$; in the three rightest columns the matrix is left-preconditioned by the Strang (S), optimal (C) and superoptimal (D) circulant preconditioners.

σ	$\mathrm{cond}(A)$	$\mathrm{cond}(S^{-1}A)$	$\mathrm{cond}(C^{-1}A)$	$\mathrm{cond}(D^{-1}A)$
2	$2.9 \cdot 10^1$	6.5	5.1	4.7
1.5	$1.3 \cdot 10^2$	$1.8 \cdot 10^1$	$1.1 \cdot 10^1$	$1.1 \cdot 10^1$
1	$2.2 \cdot 10^3$	$2.6 \cdot 10^2$	$7.1 \cdot 10^1$	$2.4 \cdot 10^2$
0.5	$3.5 \cdot 10^6$	$2.0 \cdot 10^6$	$7.2 \cdot 10^4$	$8.4 \cdot 10^5$
0.2	$4.7 \cdot 10^{12}$	$5.4 \cdot 10^{11}$	$9.0 \cdot 10^{10}$	$1.3 \cdot 10^{12}$

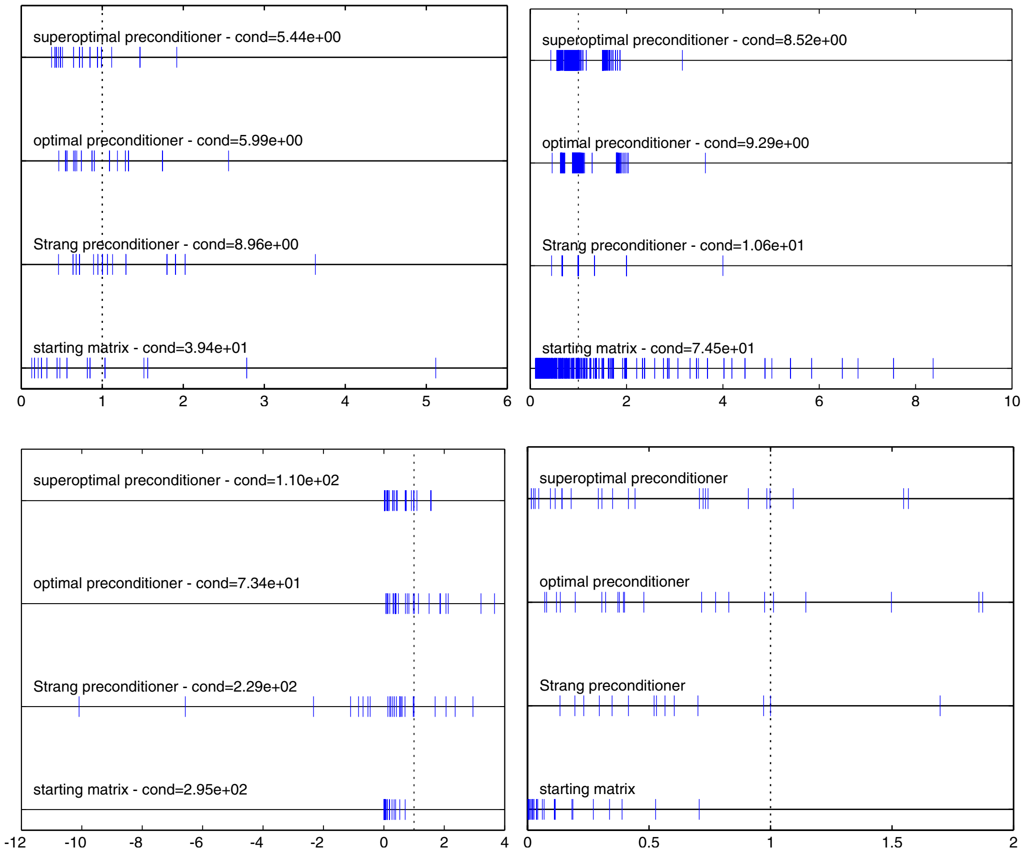

Figure 2. Effect of preconditioning on the spectrum: KMS matrix, with $\rho = 0.5$, $n_1 = n_2 = 5$ (*top left*) and $n_1 = n_2 = 20$ (*top right*); Gaussian matrix, with $\sigma_1 = \sigma_2 = 1.3$, $\theta = 1$, $n_1 = n_2 = 5$ (*complete view bottom left*, *zoomed view bottom right*).

cascade $n_1 + n_2$ one-level Toeplitz linear systems, and it is known that for one-level matrices the three preconditioners here considered are superlinear.

On the contrary, the bottom row of figure 2 shows a genuinely two-level example in which the clustering effect is rather ineffective. This test concerns the Gaussian matrix of order $n = (5, 5)$ with $\Sigma = \begin{bmatrix} 1.3 & 1 \\ 1 & 1.3 \end{bmatrix}$. This result is not surprising since, as it has been proved in [18], for every multilevel circulant preconditioner there are Toeplitz matrices for which that preconditioner is not superlinear. We note that in this particular example the Strang preconditioner is not positive definite.

To ascertain the performance of a preconditioner it is anyway necessary to look at its real effect on the convergence of an iterative method. To this end, we implemented the two-level preconditioned conjugate gradient method, which is straightforward, since the method can be stated for any positive definite self-adjoint linear transformation. In this case we applied it using the two-level formalism, that is without reordering the matrix, and employing fast algorithms for all the computations involving circulant and Toeplitz matrices by means of the two-dimensional FFT (i.e., the `fft2` function of Matlab). To highlight the numerical performance of the preconditioners, we recorded the number of iterations required to reduce the *a priori* error

$$E_k = \frac{\|\mathbf{x}^{(k)} - \mathbf{x}\|}{\|\mathbf{x}\|}$$

to 10^{-8}, where $\mathbf{x} = (1, 1, \ldots, 1)^T$ is the true solution of the linear system and $\mathbf{x}^{(k)}$ is the k-th iterate of the conjugate gradient method.

The two graphs on the left column of figure 3 concern two tensor product examples (Gaussian matrix with $\Sigma = I$ and KMS matrix with $\rho = 0.9$). Both experiments clearly confirm that the convergence of the preconditioned conjugate gradient method is superlinear, even though with different performances. These linear systems, in fact, are decomposable into one-index problems and so must conform to

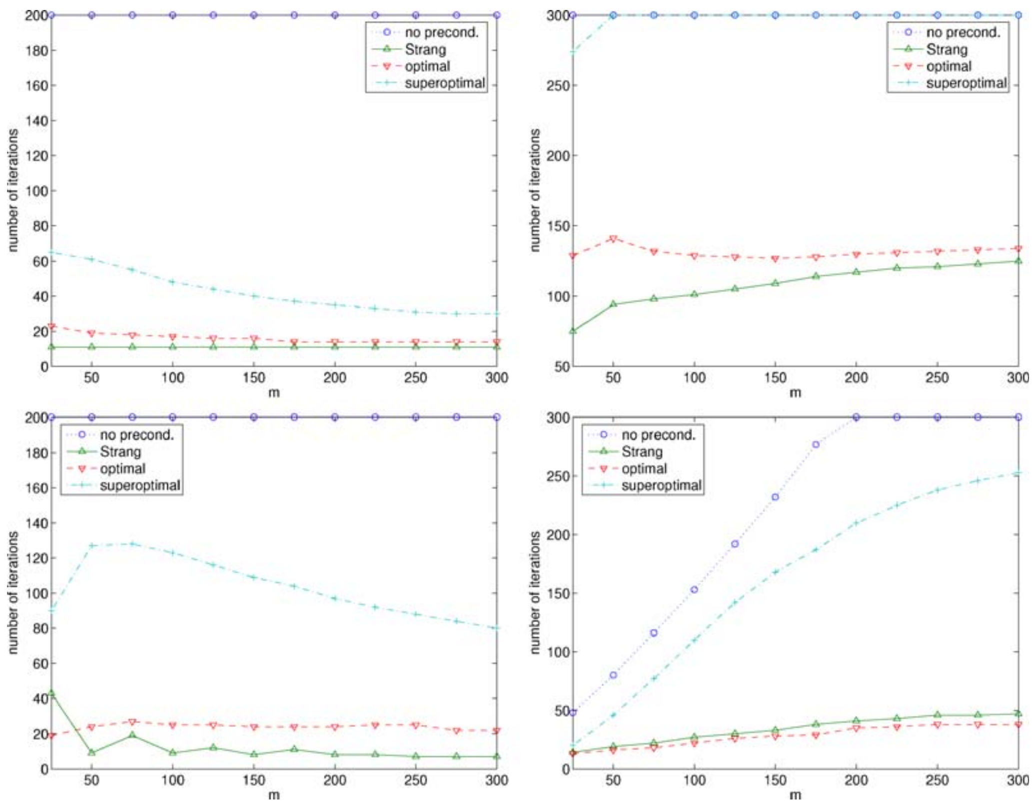

Figure 3. Number of iterations needed by the preconditioned conjugate gradient method to reduce the relative error to 10^{-8}: Gaussian matrix with $n_1 = n_2 = m = 25, 50, \ldots, 300$, $\sigma_1 = \sigma_2 = 1$, $\theta = 0$ (*top left*) and $\sigma_1 = \sigma_2 = 1.3$, $\theta = 1$ (*top right*); KMS matrix with $n_1 = n_2 = m = 25, 50, \ldots, 300$, $\rho = 0.9$ (*bottom left*) and sum of two KMS matrices with $\rho_1 = 0.5$, $\rho_2 = 0.99$ (*bottom right*).

the theory of one-level Toeplitz matrices. On the contrary, if we consider the Gaussian matrix with a non-diagonal parameter matrix Σ (top right graph in figure 3) the performance of both the Strang and the optimal preconditioners is much less impressive than before and the superoptimal preconditioner does not converge within 300 iterations.

The fourth test problem appears to be particularly interesting (bottom right graph). In such a case we consider the sum of two KMS matrices with different parameters. We recall that if a matrix is the sum of tensor products, then both the Strang and the optimal preconditioners have the same structure, while the superoptimal preconditioner does not. Hence, we expect the first two to perform much better than the last one, which is confirmed by the numerical results.

In figure 4 we investigate, by suitably varying the parameters of the two matrices, the influence of their condition numbers on the speed of convergence of the conjugate gradient method. Here, as in table 1, we observe that the preconditioners are progressively less effective as the conditioning gets worse and that the Strang and the optimal preconditioners appear more robust than the superoptimal one.

As an application, we illustrate the restoration of a blurred image [2]. The solution of this ill-conditioned problem by iterative regularization has been recently studied in [11]. We applied Gaussian blurring with $\Sigma = 0.2 \cdot I$ to a 256×256 image, shown in the top row of figure 5. The bottom row of the same figure displays the solutions obtained by six iterations of the conjugate gradient method with optimal and superoptimal preconditioning (in this case the solution obtained by Strang preconditioning is meaningless). It can be seen that both methods produce artifacts on the boundary of the images, which could be reduced significantly, for example, by imposing suitable boundary conditions on the image [12]. The boundary distortion is

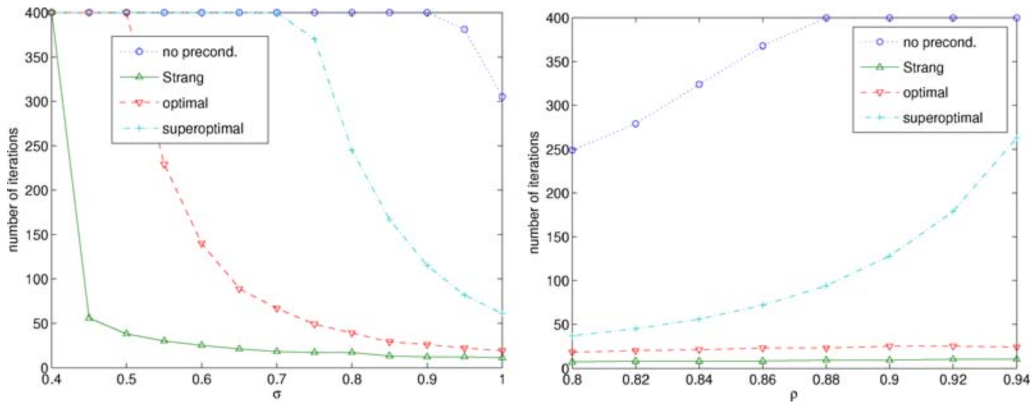

Figure 4. Effect of the variation of the condition number on the iterations needed by the preconditioned conjugate gradient method to reduce the relative error to 10^{-8}: On the *left*, Gaussian matrix with $n_1 = n_2 = 50$, $\Sigma = \sigma I$ and $\sigma = 0.4, 0.45, \ldots, 1$; on the *right*, KMS matrix with $n_1 = n_2 = 50$ and $\rho = 0.8, 0.82, \ldots, 0.94$.

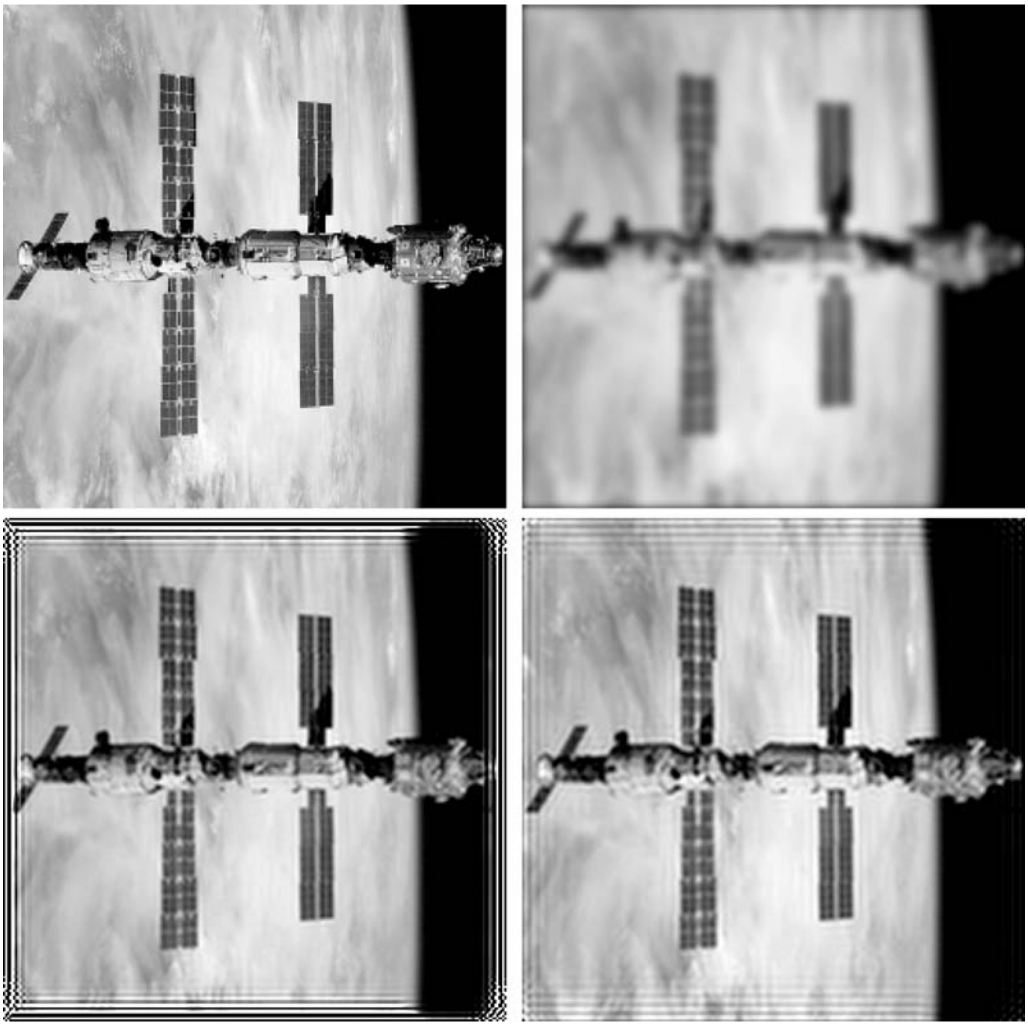

Figure 5. Original and blurred image (*top*), solution with optimal (*bottom left*) and superoptimal preconditioning (*bottom right*) after six iterations.

stronger for optimal preconditioning, but this method nevertheless produces a neater image, especially around the contour of the satellite.

Acknowledgement

This work was supported in part by MIUR, under the COFIN grant no. 2004015437, and by INdAM-GNCS.

References

[1] O. Axelsson and G. Lindskog, On the rate of convergence of the preconditioned conjugate gradient method, Numer. Math. 48(5) (1986) 499–523.

[2] M.R. Banham and A.K. Katsaggelos, Digital image restoration, IEEE Signal Process. Mag. 14 (1997) 24–41.

[3] R.H. Chan, The spectrum of a family of circulant preconditioned Toeplitz systems, SIAM J. Numer. Anal. 26 (1989) 503–506.

[4] R.H. Chan, X.Q. Jin and M.C. Yeung, The circulant operator in the Banach algebra of matrices, Linear Algebra Appl. 149 (1991) 41–53.

[5] R.H. Chan, X.Q. Jin and M.C. Yeung, The spectra of super-optimal circulant preconditioned Toeplitz systems, SIAM J. Numer. Anal. 28(3) (1991) 871–879.

[6] R.H. Chan and M.K. Ng, Conjugate gradient methods for Toeplitz systems, SIAM Rev. 38(3) (September 1996) 427–482.

[7] R.H. Chan and G. Strang, Toeplitz equations by conjugate gradients with circulant preconditioner, SIAM J. Sci. Statist. Comput. 10(1) (1989) 104–119.

[8] R.H. Chan and M.C. Yeung, Circulant preconditioners for Toeplitz matrices with positive continuous generating functions, Math. Comput. 58(197) (1992) 233–240.

[9] T.F. Chan, An optimal circulant preconditioner for Toeplitz systems, SIAM J. Sci. Statist. Comput. 9(4) (1988) 766–771.

[10] P.J. Davis, *Circulant Matrices,* Wiley, New York, 1979.

[11] F. Di Benedetto, C. Estatico and S. Serra Capizzano, Superoptimal preconditioned conjugate gradient iteration for image deblurring, SIAM J. Sci. Comput. 26(3) (2005) 1012–1035.

[12] M. Donatelli, C. Estatico and S. Serra Capizzano, Regularization of image restoration problems with anti-reflective boundary conditions, Submitted, (2004).

[13] P.C. Hansen, Deconvolution and regularization with Toeplitz matrices, Numer. Algorithms 29(4) (2002) 323–378.

[14] M. Kac, W.L. Murdock and G. Szegő, On the eigenvalues of certain Hermitian forms, J. Rat. Mech. Anal. 2 (1953) 767–800.

[15] The MathWorks, Inc., Natick, Massachusetts. *Matlab ver. 7.0*, 2004.

[16] C.V.M. van der Mee, G. Rodriguez and S. Seatzu, Fast superoptimal preconditioning of multiindex Toeplitz matrices. Submitted, (2005).

[17] C.V.M. van der Mee and S. Seatzu, A method for generating infinite positive self-adjoint test matrices and Riesz bases, SIAM J. Matrix Anal. Appl. 26(4) (2005) 1132–1149.

[18] S. Serra Capizzano and E.E. Tyrtyshnikov, Any circulant-like preconditioner for multilevel matrices is not superlinear, SIAM J. Matrix Anal. Appl. 21(2) (1999) 431–439.

[19] A. van der Sluis and H.A. van der Vorst, The rate of convergence of conjugate gradient. Numer. Math. 48(5) (1986) 543–560.

[20] Software in the Public Interest, Inc. *Debian/GNU Linux 3.1 (Sarge)*, 2005. http://www.debian.org/.

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

[22] V.V. Strela and E.E. Tyrtyshnikov, Which circulant preconditioner is better? Math. Comput. 65(213) (1996) 137–150.

[23] M. Tismenetsky, A decomposition of Toeplitz matrices and optimal circulant preconditioning, Linear Algebra Appl. 154/156 (1991) 105–121.

[24] W.F. Trench, Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices, SIAM J. Matrix Anal. Appl. 10 (1989) 135–146.

[25] E.E. Tyrtyshnikov, Optimal and superoptimal circulant preconditioners, SIAM J. Matrix Anal. Appl. 13(2) (1992) 459–473.