SYMMETRIC CAUCHY-LIKE PRECONDITIONERS FOR THE REGULARIZED SOLUTION OF 1-D ILL-POSED PROBLEMS *

MISHA E. KILMER†

Abstract. The discretization of integral equations can lead to systems involving symmetric Toeplitz matrices. We describe a preconditioning technique for the regularized solution of the related discrete ill-posed problem. We use discrete sine transforms to transform the system to one involving a Cauchy-like matrix. Based on the approach of Kilmer and O'Leary, the preconditioner is a symmetric, rank m^* approximation to the Cauchy-like matrix augmented by the identity. We shall show that if the kernel of the integral equation is smooth then the preconditioned matrix has two desirable properties; namely, the largest m^* magnitude eigenvalues are clustered around and bounded below by one, and that small magnitude eigenvalues remain small. We also show that the initialization cost is less than the initialization cost for the preconditioner introduced by Kilmer and O'Leary. Further, we describe a method for applying the preconditioner in $O((n+1)\lg(n+1))$ operations when $n+1$ is a power of 2, and describe a variant of the MINRES algorithm to solve the symmetrically preconditioned problem. The preconditioned method is tested on two examples.

Key words. Regularization, ill-posed problems, Toeplitz, Cauchy-like, preconditioner, conjugate gradient, minimal residual, normal equations, image processing, deblurring

AMS(MOS) subject classifications. 65R20, 45L10, 94A12

December 4, 1997

1. Introduction.

In many applications, a one-dimensional first kind integral equation of the form

$$\int_{\beta_1}^{\beta_u} t(\alpha, \beta) f(\beta) d\beta = g(\alpha)$$

is used to model the output response of an instrument or system to input data. These one-dimensional integral equations are often solved using a discretization that results in a least squares problems or a linear system. The corresponding discrete, noisy system is of the form

$$Tf = g = \hat{g} + e \tag{1}$$

where T is symmetric and Toeplitz, $\hat{g}$ represents the noise free data, e represents noise, and g is the actual measured data. (We shall assume that T is $n \times n$, but note that the preconditioning scheme to be introduced in this paper could be adjusted for the rectangular case as described in [16]).

Given only T and the noisy data g, one would like to approximate the exact solution $\hat{f}$ to the noise-free problem $T\hat{f} = \hat{g}$. However, since the continuous problem is ill-posed, the matrix T is ill-conditioned. It is easy to show that the exact solution to (1) is hopelessly contaminated by noise since the small singular values magnify the noise components in g. Therefore some form of *regularization* needs to be used to determine an approximate solution to $\hat{f}$.

Since a number of computations involve discrete sine transforms, we shall assume that $n+1$ is a power of 2 so that the related operations counts can be written in terms

---

* This work was supported by the National Science Foundation under Grant CCR 95-03126.

† Applied Mathematics Program, University of Maryland, College Park, MD 20742 (mkilmer@cs.umd.edu)

of $O(N \lg N)$ where $N = n + 1$. If $n + 1$ is not a power of two, we can augment the matrix T by an identity matrix of appropriate size, so that now T is Toeplitz block with 2 Toeplitz matrices on the block diagonal. However, the displacement rank of the augmented matrix will remain the same as the original matrix (≤ 4).

Iterative Krylov subspace methods can be used as regularization techniques [16, 15, 8, 7]. They can be particularly efficient on problems involving Toeplitz matrices since multiplication of a Toeplitz matrix times a vector can be done quickly (see [14]). However, algorithms such as CGLS (conjugate gradient for least squares) and MINRES (minimal residual, [17]) can be slow to converge to a regularized solution of (1). Therefore, we look for preconditioners which will speed convergence to a regularized solution while filtering noise in early iterates.

As in [16], the idea is to use the rank revealing properties of a factorization of T to develop a preconditioner. However, when the necessary pivoting is incorporated into the fast or super-fast LDU factorization algorithms for Toeplitz matrices, these methods can become too expensive for our purposes, often requiring $O(n^3)$ operations to factor an $n \times n$ Toeplitz matrix [18, 6, 1]. To circumvent this problem, in [16] we advocated transforming T to a Cauchy-like matrix C using a particular unitary transformation. However, if T was symmetric, C in that case was no longer symmetric. This lack of symmetry prevents one from applying such algorithms as MINRES or MR-II [7] to solve the transformed system — rather, an algorithm such as CGLS, which requires roughly twice as much work per iteration, must be used.

To overcome this difficulty, we note that a Toeplitz matrix T is orthogonally related to a partially reconstructible (PR) Cauchy-like matrix C via discrete trigonometric transforms [14]. One such relation considered here uses discrete sine transforms and allows us to preserve symmetry and to determine a fast algorithm for applying the preconditioner. If the kernel of the integral equation is smooth, then we observe that C has the property that its largest magnitude elements lie in its leading principal submatrix of dimension $m^* \times m^*$ (see §6). This implies that no pivoting is needed to partially factor C as it was in [16]. Since the $m^* \times m^*$ submatrix is itself a PR Cauchy-like matrix, it is possible to compute the LDU factorization in $O((m^*)^2)$ operations; thus our method requires less initialization overhead than the method in [16].

This paper is organized as follows. In §2, we define a partially reconstructible Cauchy-like matrix and give some of its properties. In §3, we give some background on regularization and preconditioning in the context of regularization. We introduce our preconditioner in §4 and give theoretical results in §5. We discuss the properties of the transformation in §6 which allow us by-pass the pivoting stage. Algorithmic issues are addressed in §7 and a preconditioned variant of MINRES is given. Numerical results are the subject of §8 and conclusions are given in §9.

2. Partially reconstructible Cauchy-like matrices.

A *partially reconstructible* Cauchy-like matrix can be represented in the form:

$$C_{ij} = \begin{cases} \frac{\tilde{a}_i^T \tilde{b}_j}{\omega_i - \omega_j} & i \neq j \\ c_i & i = j \end{cases}$$

where c_i denote the diagonals of the matrix C, and $\tilde{a}_i, \tilde{b}_j \in \mathcal{C}^{\ell \times n}$. The matrices

$$\tilde{A} = \begin{bmatrix} \tilde{a}_1^T \\ \vdots \\ \tilde{a}_n^T \end{bmatrix} \text{ and } \tilde{B} = \begin{bmatrix} \tilde{b}_1^T \\ \vdots \\ \tilde{b}_n^T \end{bmatrix}$$

are called the *generators* of the matrix and ℓ is called the *displacement rank.* Note that the entries of C are completely characterized in terms of the generators, the n numbers ω_i, and the n diagonal entries of the matrix.

The following property shows how a Toeplitz matrix can be transformed into a partially reconstructible Cauchy-like matrix [12, 14]:

PROPERTY 1. *Every Toeplitz matrix T satisfies an equation of the form*

$$HT - TH = AB^T \tag{2}$$

where where $A \in \mathcal{C}^{n \times \ell}$, $B \in \mathcal{C}^{n \times \ell}$, $1 \leq \ell \leq 4$, and

$$H = \frac{1}{2}\text{tridiag}(1, 0, 1).$$

The Toeplitz matrix T is orthogonally related to a partially reconstructible Cauchy-like matrix

$$C = STS$$

that satisfies the displacement equation

$$DC - CD = \tilde{A}\tilde{B}^T, \ \ \tilde{A} = SA, \ \ \tilde{B} = SB, \tag{3}$$

where

$$D = \text{diag}\left(\cos\left(\frac{\pi}{n+1}\right), \cos\left(\frac{2\pi}{n+1}\right), \ldots, \cos\left(\frac{n\pi}{n+1}\right)\right)$$

and S is the normalized discrete sine transform matrix

$$S = \sqrt{\frac{2}{n+1}}\left[\sin\left(\frac{kj\pi}{n+1}\right)\right]_{k,j=1}^{n}.$$

The authors of [12] give an explicit formula for computing the diagonal entries of C which are unspecified by (3). Alternately, these entries can be computed by diagonalizing the corresponding T. Chan-type preconditioner described in [14]. Fortunately, since we have assumed $N = n+1$ is a power of 2, this can be done quickly by means of fast sine (and cosine in the case of [12]) transforms in $O(N \lg N)$ operations. Note that the generators of T are readily determined from (2) and the generators of C can be determined with fast sine transforms.

The next property gives some insight into how matrix-vector multiplications might be computed.

PROPERTY 2. *Let C_0 be the Cauchy matrix*

$$(C_0)_{ij} = \begin{cases} \frac{1}{\omega_i - \omega_j}, & ii \neq j \\ 0 & i = j \end{cases}$$

Following [2], we observe

$$C = (\sum_{i=1}^{\ell} \mathrm{diag}(\tilde{A}^{(i)}) C_0 \mathrm{diag}(\tilde{B}^{(i)}) + \mathrm{diag}(c), \tag{4}$$

where the superscript on $\tilde{A}$ and $\tilde{B}$ denotes the ith column of the generators, c denotes the vector with components c_i, and diag$(\cdot)$ means the diagonal matrix formed by placing the vector argument along the diagonal.

We will make use of (4) to determine a fast algorithm for applying the preconditioner (see §7).

There are two other properties of Cauchy-like matrices which we will be able to exploit; namely, that the inverse of a PR Cauchy-like matrix is PR Cauchy-like and that the leading principal submatrix of a PR Cauchy-like matrix is PR Cauchy-like. Both of these two properties can be observed by appropriately manipulating (3). From (3) we can also deduce that the generators X and W for C^{-1} can be found by solving

$$CX = \tilde{A}, \;\; W^T C = \tilde{B}^T. \tag{5}$$

3. Regularization and preconditioning. Throughout this paper, we will make the following four assumptions:

1. The matrix T has been normalized so that its largest eigenvalue is of order 1.
2. The uncontaminated data vector $\hat{g}$ satisfies the discrete Picard condition; i.e., the spectral coefficients of $\hat{g}$ decay in absolute value faster than the singular values [20, 10].
3. The additive noise is zero-mean white Gaussian. In this case, the components of the error e are independent random variables normally distributed with mean zero and variance ϵ^2.
4. The *noise level*, $\frac{\|e\|_2}{\|\hat{g}\|_2}$, is strictly less than one.

Since T is symmetric, let $T = V\Lambda V^T$ be the eigendecomposition of T, where the entries in the diagonal matrix Λ are the eigenvalues $\lambda_i, i = 1, \ldots, n$ with $|\lambda_1| \geq |\lambda_2| \ldots \geq |\lambda_n|$. The spectral coefficients of the exact data $\hat{g}$ and noise e are $\zeta = V^T\hat{g}$ and $\eta = V^T e$, respectively.

It is easy to show that the exact solution to (1) is given in spectral coordinates by

$$f = \sum_{i=1}^{n} \frac{\zeta_i + \eta_i}{\lambda_i} v_i, \tag{6}$$

where v_i denotes the ith column of V.

Under the white noise assumption, $|\eta_i| \approx \epsilon, i = 1, \ldots r$ so that the noise coefficients are roughly constant, while the discrete Picard condition tells us that the ζ_i go to zero at least as fast as the singular values σ_i. Thus, components for which ζ_i is of the same order or less than η_i are obscured by noise.

By assumptions 2 and 4, there exists $\bar{m} > 0$ such that for all $i > \bar{m}$, the ζ_i are indeed indistinguishable from the η_i. Further, there exists $0 < m^* \leq \bar{m}$ such that for $i > m^*$ it is never the case that $|\zeta_i| \gg |\eta_i|$. As in [16], we therefore choose to partition the columns of V into bases for the *upper, lower,* and *transition* subspaces as follows. We say that the *upper* subspace is the space spanned by the first m^* columns of V. Hence the upper subspace corresponds to the largest m^* singular values. The *lower* subspace is the space spanned by the last $n - \bar{m}$ columns for V;

i.e. those columns of V corresponding to the smallest singular values. Finally, the *transition* subspace is the space spanned by the remaining $\bar{m}-m^*$ columns of V. Since these columns correspond to the mid-range singular values, the transition subspace is generally difficult to resolve unless there is a gap in the singular value spectrum.

The exact solution to the noise-free least squares problem can also be expanded in terms of the eigendecomposition of T:

$$\hat{f} = \sum_{i=1}^{n} \frac{\zeta_i}{\lambda_i} v_i. \tag{7}$$

Comparing (7) with (6) we see that f resembles $\hat{f}$ on the upper subspace, yet our assumptions also require that f and $\hat{f}$ differ greatly in the magnitude of their components in the lower subspace; the components of $\hat{f}$ in the lower subspace are small while the components of f in the lower subspace are large and increase in magnitude as i approaches n. We would therefore like our regularization method to produce a regularized solution with small components in the lower subspace and which resembles $\hat{f}$ in the upper subspace. Fortunately, Krylov subspace methods such as MINRES and CGLS tend to produce this type of solution, with the iteration index taking the role of the regularization parameter. To speed convergence to a regularized solution, we must develop a preconditioner which clusters the first m^* eigenvalues (in absolute value) around one (see [19]); however, to keep the preconditioner from mixing noise into early iterates, we also want the small singular values, and with them, the lower subspace, to be unchanged.

4. The preconditioner. Let $C = STS$ be the partially reconstructible (PR) Cauchy-like matrix corresponding to the Toeplitz matrix T. Solving $Tf = g$ is equivalent to solving

$$CSf = Sg,$$

Let $T = V\Lambda V^T$ be the singular value decomposition of T. Since S is an orthogonal matrix,

$$C = SV\Lambda V^T S^T,$$

where $S = S^T$, so that C and T have the same eigenvalues and there is no mixing of the upper and lower subspaces by changing to the new coordinate system.

In [16], in order to determine the preconditioner one first had to perform a partial factorization of the corresponding Cauchy-like matrix in order to permute the largest magnitude components of C to the leading principal submatrix. We show in §6 that as a property of the transformation, the leading principal submatrix already contains the large magnitude entries. Therefore we save the cost of performing the partial factorization.

Setting $z = Sf$ and $g = Sg$, the problem $Tf = g$ is equivalent to

$$Cy = z.$$

If we desire to use CGLS, we would choose, as in [16], a preconditioner for the left so that

$$M^{-1}Cy = M^{-1}z. \tag{8}$$

If T, and hence C, is symmetric, however, we may want to find a symmetric preconditioner M and apply MINRES or MR-II to the symmetrically preconditioned system

$$M^{-1/2}CM^{-1/2}\hat{y} = M^{-1/2}z \tag{9}$$

where $\hat{y} = M^{1/2}y$. It turns out that in this case, both MINRES and MR-II for the symmetrically preconditioned problem can be written in terms of the matrix M^{-1} rather than $M^{-1/2}$ (see §7.4).

Writing C in block form we have

$$\begin{bmatrix} C_1 & C_2 \\ C_2^T & C_4 \end{bmatrix},$$

where C_1 is $m^* \times m^*$. The permutation ensures that C_1 is well-conditioned having the largest magnitude elements of C. The preconditioner M is then defined as in [16]:

$$M = \begin{bmatrix} C_1 & 0 \\ 0 & I \end{bmatrix}.$$

5. Properties of the Preconditioner. Since M is defined in the same way as in [16], the theory in [16] tells us that the left preconditioned matrix has the desired properties; namely, that the largest m^* singular values are clustered around 1, while the lower subspace, and the small singular values, remain relatively untouched. Therefore we expect CGLS to give reasonable regularized solutions after only a relatively small number of iterations.

However, if C is symmetric, we may want to apply MINRES or MR-II to the symmetrically preconditioned problem (9). Thus, we need to show that the largest magnitude eigenvalues of $M^{-1/2}CM^{-1/2}$ are clustered around one while the smallest magnitude eigenvalues remain small. If T is symmetric than so are C, M, and $M^{-1/2}CM^{-1/2}$. Since $M^{-1/2}CM^{-1/2}$ is symmetric, the absolute values of its eigenvalues are precisely its singular values. Similarly, the absolute values of the eigenvalues of C are its singular values. Since we are interested in clustering eigenvalues by magnitude, it will be convenient to show the appropriate clustering results for the singular values of $M^{-1/2}CM^{-1/2}$ instead.

Observe that $M^{-1}C$ and $M^{-1/2}CM^{-1/2}$ have the same eigenvalues since the two matrices are related via a similarity transform. Define $\hat{s} = \max\{\|C_1^{-1}C_2\|_\infty, \|C_2\|_\infty\}$. It will be convenient to decompose the matrix $(M^{-1/2}CM^{-1/2})^2$ as

$$\begin{bmatrix} I \\ C_2^T C_1^{-1/2} \end{bmatrix} \left[I, C_1^{-1/2}C_2\right] + \begin{bmatrix} C_1^{-1/2}C_2 \\ C_4 \end{bmatrix} \left[C_2^T C_1^{-1/2}, C_4\right] = E_{1M} + E_{2M}. \tag{10}$$

Theorem 5.1. *$|\lambda_i(M^{-1/2}CM^{-1/2})|, i = 1, \ldots, m^*$ are bounded below by 1 and above by $1 + \hat{s}$.*

Proof: Proceed as in Theorem 3.1 of [16] to deduce that m^* of the singular values of $M^{-1/2}CM^{-1/2}$ are bounded below by 1. The upper bound comes from applying Gershgorin's Theorem to $M^{-1}C$ and using the similarity transform. □

To show that the small magnitude eigenvalues remain small, decompose C^*C as

$$\begin{bmatrix} C_1 \\ C_2^T \end{bmatrix} [C_1, C_2] + \begin{bmatrix} C_2 \\ C_4 \end{bmatrix} [C_2^T, C_4] = E_{1C} + E_{2C}. \tag{11}$$

We have the following theorem:

THEOREM 5.2. *Let* $c_{m^*} = \max\{1, \sqrt{\frac{1}{\sigma_{m^*}(C_1)}}\}$*. Then the* (m^*+i)*th largest magnitude eigenvalue of* $M^{-1/2}CM^{-1/2}$ *lies in the interval* $[0, c_{m^*}\sqrt{\sigma_i(E_{2C})}]$ *and the* (m^*+i)*th largest magnitude eigenvalue of* C *lies in the interval* $[0, \sqrt{\sigma_i(E_{2C})}]$.

Proof: Proceeding as in Theorem 3.3 of [16], we can show

$$\lambda_{i+m^*}(C^2) \leq \lambda_i(E_{2C}), i = 1, \ldots, m^*$$

and

$$\lambda_{i+m^*}((M^{-1/2}CM^{-1/2})^2) \leq \lambda_i(E_{2M}), i = 1, \ldots, m^*.$$

Now $E_{2M} = M^{-1/2}E_{2C}M^{-1/2}$. Thus two applications of Theorem 3.3.16d to the right hand side of the above equation yields

$$\sigma^2_{i+m^*}(M^{-1/2}CM^{-1/2}) \leq \sigma_1(M^{-1})\sigma_i(E_{2C}), i = 1, \ldots, m^*.$$

We also have

$$\sigma^2_{i+m^*}(C) \leq \sigma_i(E_{2C}), i = 1, \ldots, m^*$$

The proof is completed by taking square roots. □

6. Properties of the Transformation. These theorems show that the preconditioner will be effective if C_1 is well-conditioned and if the row sums of $C_1^{-1}C_2$ and E_{2C} are small. We now discuss to what extent we expect these conditions to hold for integral equation discretizations. We shall assume C is symmetric.

Let $\tilde{A}$ and $\tilde{B}$ be the generators of C. From Property 1 we have

$$(C)_{ij} = \begin{cases} \frac{\tilde{a}_i^T \tilde{b}_j}{\omega_i - \omega_j}, & i \neq j \\ c_j, & \text{otherwise} \end{cases},$$

where the values c_j denote the diagonal entries of C. The values c_j for a symmetric matrix C are the entries of the diagonal matrix SC_SS, where C_S is the Chan-type preconditioner in [14]. Using this relationship, an exact formula can be determined for computing the c_j (see [14]): Let $t_i, i = 0, \ldots, n-1$ denote the diagonals of T and define $s_{jk} = \sin(jk\pi/(n+1))$, $t_n = 0$, and

$$r_k = \begin{cases} t_0 - \frac{n-2}{n+1}t_2, & k = 1 \\ \frac{n-k+3}{n+1}t_{k-1} - \frac{n-k-1}{n+1}t_{k+1}, & k > 1 \end{cases}.$$

Then

$$c_j = \frac{1}{\sin(j\pi/(n+1))}\sum_{k=1}^{n} r_k s_{jk}. \tag{12}$$

Now Heinig and Bojanczyk [12] show that the off-diagonal elements C_{ij} for which $i+j$ is odd are 0 while if $i+j$ is even, we have

$$C_{ij} = \frac{1}{\cos(\frac{i\pi}{n+1}) - \cos(\frac{j\pi}{n+1})}(\frac{2}{n+1})\left(\sin(\frac{j\pi}{n+1})\sum_{k=1}^{n} s_{ik}t_k - \sin(\frac{i\pi}{n+1})\sum_{k=1}^{n} s_{jk}t_k\right).$$

Therefore, for $i \neq j$, $i+j$ even,

$$|C_{ij}| \leq \frac{1}{\cos(\frac{i\pi}{n+1}) - \cos(\frac{j\pi}{n+1})}(\frac{2}{n+1}) \left(|\sum_{k=1}^{n} s_{ik}t_k| + |\sum_{k=1}^{n} s_{jk}t_k| \right).$$

From (12),

$$|C_{ii}| = \frac{1}{|\sin(\frac{i\pi}{n+1})|} |\sum_{k=1}^{n} s_{ik}r_k|.$$

Now $\sum_{k=1}^{n} s_{ik}t_k$ is the ith coefficient of the (unnormalized) discrete sine transform (DST) of the vector $v = [0, t_1, t_2, \ldots, t_{n-1}, 0]$. But this is, up to a factor $2\sqrt{-1}$, the ith coefficient of the discrete Fourier transform of the vector v^e, the odd-extension of v about $v_{n+1} = 0$ [13]. If the kernel of the integral equation is smooth, then the Fourier coefficients tend to decrease in magnitude quickly as i approaches n. Thus, the DST coefficients v_i of v tend to decrease in i. Since $|\sum_{k=1}^{n} s_{ik}t_k| \leq \sum_{k=1}^{n} |t_k|$, if $t_k < 1$, this implies many of the DST coefficients of v are small. Likewise, as the r_k correspond to a linear combination of the t_k, the magnitude of the DST coefficients of the vector r decrease with i. Since

$$\sum_{k=1}^{n} |r_k| \leq |t_0| + \frac{n-2}{n+1}|t_2| + \left(\sum_{k=2}^{n} \frac{n-k+3}{n+1}|t_{k-1}| + \frac{n-k-1}{n+1}|t_{k+1}| \right),$$

the magnitude of the DST coefficients of r get small as i increases.

Next, consider $1/\left(\cos(\frac{i\pi}{n+1}) - \cos(\frac{j\pi}{n+1})\right)$ as a function of i and j, for $i \neq j$, $i+j$ even. Clearly this expression decays rapidly away from the diagonal when n is large (see Figure 1 for an illustration). Recalling that the DST coefficients of v become small as i and j increase, this means that the trailing submatrix of C and the upper right and lower left corners of C contain the smallest components of the matrix. Figure 2 plots $\frac{2}{n+1}(|\sum_{k=1}^{n} s_{ik}t_k| + |\sum_{k=1}^{n} s_{jk}t_k|)$ for $j = 1, \ldots, n$ for a few fixed values of i for the vector t defined in Example 2. (The spy plot in Figure 9 shows the actual magnitudes of the entries of C for Example 2.)

Now consider $1/\sin(i\pi/(n+1))$. For sufficiently large n, this quantity is large for small i, decays quickly toward 1 as i increases toward $(n+1)/2$, and becomes large as i approaches n (see Figure 3). A plot of $|\sum_{k=1}^{n} r_k s_{ik}|$ for the r_k of Example 2 (see Fig 4) is included for comparison. Since the DST coefficients of r become small as i increases, clearly the diagonal elements of C are large only for the first few values of i. Hence, there exists a leading principal submatrix of C which contains most of the large magnitude elements of C.

7. Algorithmic Issues. For a symmetric matrix T, our algorithm is as follows:

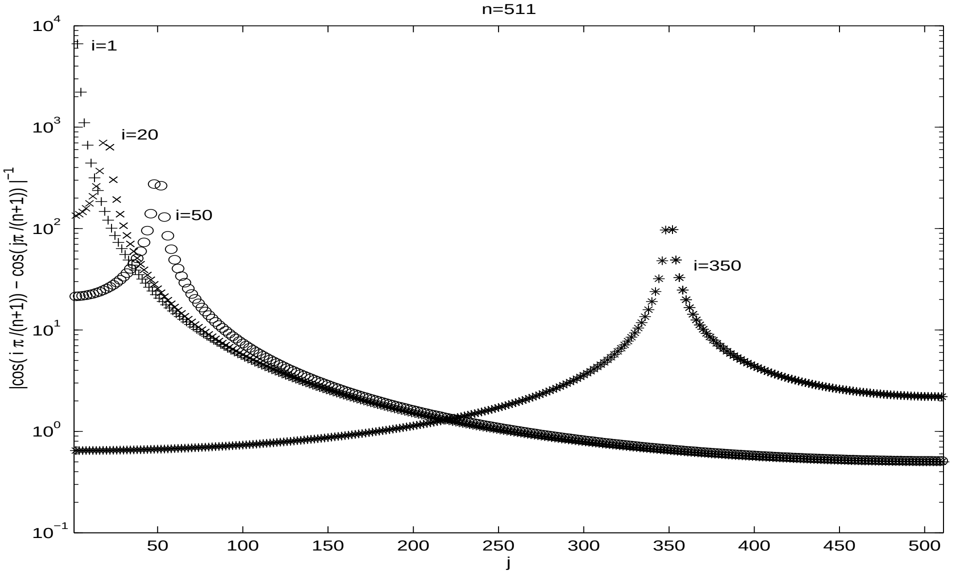

FIG. 1. *Plot of* $\frac{1}{|\cos(i\pi/(n+1))-\cos(j\pi/(n+1))|}$ *as a function of* j *for fixed values of* i *for* $n = 511$.

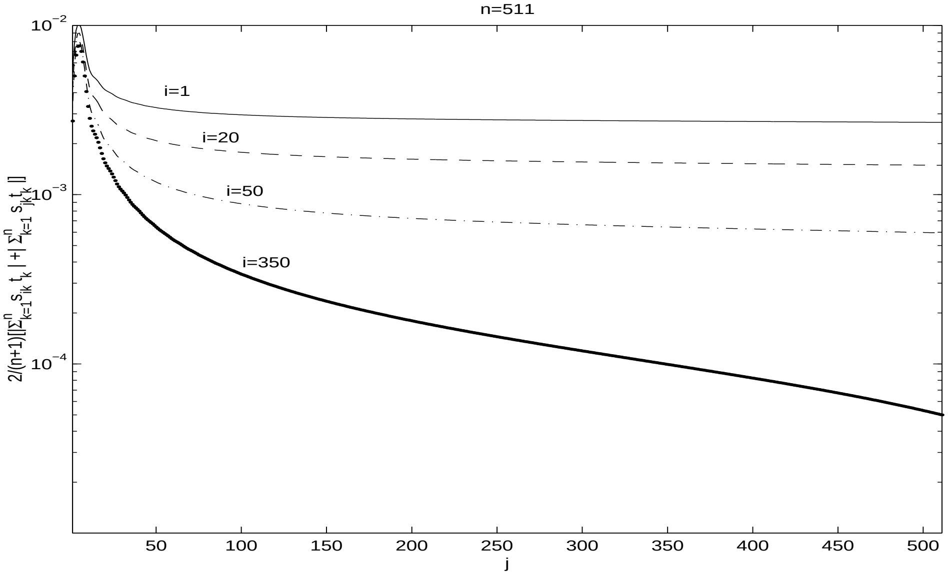

FIG. 2. *Plot of* $\frac{2}{n+1}\left(|\sum_{k=1}^{n} s_{ik}t_k| + |\sum_{k=1}^{n} s_{jk}t_k|\right)$ *for* $j = 1, \ldots, 511$, *for a few fixed values of* i *for the vector* t *in Example 2.*

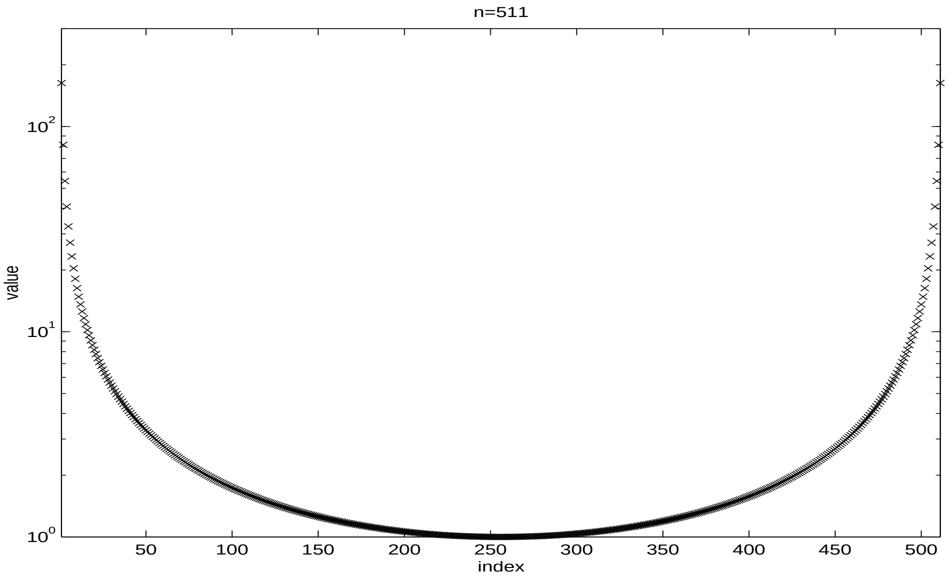

FIG. 3. *Plot of* $\frac{1}{|sin(i\pi/(n+1))|}$ *for* $n = 511$.

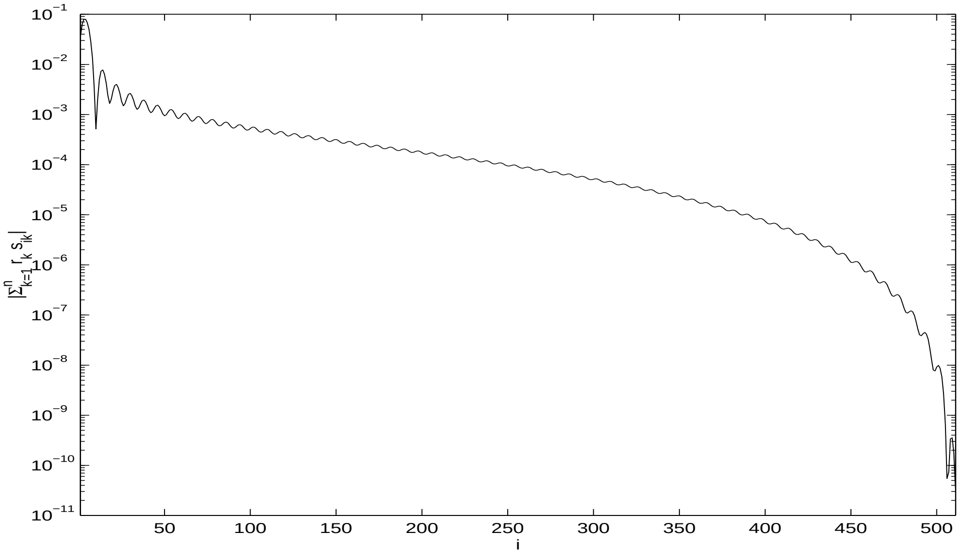

FIG. 4. *Plot of* $|\sum_{k=1}^{n} r_k s_{ik}|$ *for the* r_k *of Example 2.*

```
Algorithm 3: Solving $Tf = g$ for symmetric $T$
1. Compute the generators $\tilde{A}$ and $\tilde{B}$ for the matrix $C = STS$.
2. Compute the diagonal entries of $C$ according to (12).
3. Determine the index $m^*$ to define the size of $C_1$.
4. Compute the generators $A_1$ and $B_1$ of $C_1$.
5. Factor $C_1$, and use forward and back substitution to determine the generators $X$
   and $W$ of $C_1^{-1}$.
6. Solve $C_1^{-1} C_1 = I$ for the diagonal entries of $C_1^{-1}$ ( see §7.1).
7. Compute an approximate solution $\tilde{y}$ to $M^{-1/2} C M^{-1/2} (M^{1/2} y) = M^{-1/2} z$ using a
   few steps of MINRES or MR-II.
8. The approximate solution in the original coordinate system is
   $f = S\tilde{y}$.
```

We note that we do not address the problem of determining when it is best to stop iterating to get a good solution. The interested reader is referred to [8] for a discussion of how the L-curve method can be used to determine an appropriate regularization parameter for MR-II and to [7] for how Morozov's discrepancy principle can be used to find a regularization parameter for MINRES (see also [11]).

7.1. Determining M^{-1}. Since C satisfies the displacement equation (3), it follows that C_1 is a partially reconstructible Cauchy-like matrix satisfying

$$\Omega_1 C_1 - C_1 \Omega_1 = A_1 B_1^T$$

where Ω_1 is the leading $m^* \times m^*$ principal submatrix of D in (3) and A_1 and B_1 contain the first m^* columns of $\tilde{A}$ and $\tilde{B}$, respectively.

Thus, the matrix C_1^{-1} is partially reconstructible, with off diagonal entries given by

$$(C_1^{-1})_{ij} = -\frac{x_i^T w_j}{\omega_i - \omega_j}, i \neq j$$

where the vectors x_i^T and w_j^T are rows of X_1 and W_1 defined as

$$C_1 X_1 = A_1, \; W_1^T C_1 = B_1^T.$$

Computing X_1 and W_1 costs $O((m^*)^2)$ operations, given the factorization of C_1 and the matrices A_1 and B_1. To get A_1 and B_1, we simply need $\tilde{A}$ and $\tilde{B}$, which we can obtain from the generators of T in $O(N \lg N)$ operations using the fast sine transform.

Since C_1^{-1} is partially reconstructible, its diagonal entries $\hat{c}_j$ cannot be determined from its displacement equation. However, the $\hat{c}_j$, can be computed from the simple relation $C_1^{-1} C_1 = I$ in $O((m^*)^2)$ operations since we know the off-diagonal elements of C_1^{-1} and all the elements of C_1. The total initialization cost of the preconditioner, which includes the time to determine $\tilde{A}$ and $\tilde{B}$ and solving for X_1 and W_1 is therefore $O((m^*)^2 + N \lg N)$ operations.

7.2. Applying the preconditioner. Since we are using a different transformation to Cauchy-like than that used in [16], we need a different method for quickly applying the preconditioner. Let v be a vector of length m^*, and assume that the permutation matrix is the identity. Now from (4) applied to C_1^{-1}, we see matrix vector products with C_1^{-1} can be formed as

$$C_1^{-1}v = \left(\sum_{i=1}^{\ell} -X_1^{(i)} \cdot (\hat{C}_0(W_1^{(i)} \cdot v))\right) + \mathrm{diag}(\hat{c})v$$

where $\hat{C}_0$ is the $m^* \times m^*$ leading principle submatrix of C_0, $X_1^{(i)}$, $W_1^{(i)}$ are the ith columns of X_1 and W_1, $\hat{c}$ is the vector with components $\hat{c}_i$, and $\cdot$ denotes componentwise multiplication. Computing a matrix-vector product with a Cauchy matrix of the form C_0 is known as Trummer's problem. Suppose we extend the vector $u = W_1^{(i)} \cdot v$ to n dimensions and replace $\hat{C}_0$ with C_0. Then it is possible to compute $C_0 u$ in $O(N \lg N)$ operations using fast sine and cosine transformations via a variant of the algorithm of Gerasoulis *et al* [4] for solving Trummer's problem, which we now describe.

Let the polynomials $h(x)$ and $s(x)$ be defined according to

$$\frac{h(x)}{s(x)} = \sum_{i=1}^{n} \frac{u_i}{x - \omega_i}.$$

Note that $h(\omega_i) = u_i s'(\omega_i)$. Now Gerasoulis in [3] shows that the jth component of $C_0 u$ can be determined through an appropriate evaluation of polynomials:

$$z_j = (h'(\omega_j) - \frac{1}{2} u_j s''(\omega_j))/s'(\omega_j). \tag{13}$$

Now $s(x) = \prod_{i=1}^{n}(x - \omega_i)$. But the ω_i are just the roots of the Chebyshev polynomial of the second kind of degree $n+1$, denoted $U_{n+1}(x)$. Thus, $s(x)$ can be written in terms of an nth degree polynomial of the second kind as $s(x) = 2^{-n}U_n(x)$ [3]. Using this formula for $s(x)$, it is easy to show

$$s'(\omega_j) = 2^{-n}(-1)^{j+1}(n+1)/\sin^2(j\pi/(n+1)) \tag{14}$$

and $s''(\omega_j)/s'(\omega_j) = 3\cos(j\pi/(n+1))/\sin^2(j\pi/(n+1))$, so (13) reduces to

$$z_j = h'(\omega_j)/s'(\omega_j) - \frac{3\cos(j\pi/(n+1))}{2\sin^2(j\pi/(n+1))} u_j \tag{15}$$

and it remains to find an expression for $h'(\omega_j)/s'(\omega_j)$.

To determine $h'(x)$, we first set $h(x) = \sum_{k=1}^{n} a_k U_{k-1}(x)$. The coefficients a_k can now be found using the fact that $h(\omega_i) = u_i s'(\omega_i)$, $i = 1, \ldots, n$. From $h(\omega_i) = u_i s'(\omega_i)$, close inspection shows that

$$a_k = 2^{-n}(n+1)\frac{2}{n+1}\sum_{i=1}^{n} \frac{\sin(ki\pi/(n+1))}{\sin(i\pi/(n+1))} u_i. \tag{16}$$

Since the a_k are known, we can use the relation $h(x) = \sum_{k=1}^{n} a_k U_{k-1}(x)$ to determine $h'(\omega_j)$. We obtain

$$\begin{aligned} h'(\omega_j) \; &= \; \frac{1}{\sin^2(j\pi/(n+1))} \sum_{k=1}^{n} -a_k k \cos(kj\pi/(n+1)) \\ &\quad + \frac{\cos(j\pi/(n+1))}{\sin^3(j\pi/(n+1))} \sum_{k=1}^{n} a_k \sin(kj\pi/(n+1)). \end{aligned} \tag{17}$$

Next, substitute (16) for a_k and factor the constants $2^{-n}(n+1)$ out in front of the sum. Dividing this expression by (14) and setting $y_i = \frac{(-1)^{i+1}}{\sin(i\pi/(n+1))} u_i$, we obtain

$$
\begin{aligned}
h'(\omega_j)/s'(\omega_j) &= \frac{\cos(j\pi/(n+1))}{\sin^3(j\pi/(n+1))} y_j \\
&\quad - \frac{2(-1)^{j+1}}{n+1} \sum_{k=1}^{n} k \cos\left(\frac{kj\pi}{n+1}\right) \sum_{i=1}^{n} \sin\left(\frac{ij\pi}{n+1}\right) u_i . \qquad (18)
\end{aligned}
$$

Together with (15), this means that the components of z_j can be computed simultaneously by means of fast $O(N \lg N)$ sine and cosine transforms of dimension n. This observation leads us to develop an algorithm for computing $C_1^{-1} v$ which costs only $O(N \lg N)$ operations:

Algorithm 4: Forming $\hat{z} \leftarrow C_1^{-1} v$

```
Set $\hat{v} = 0$.
For $j = 1, \ldots, \ell$, do
    1. Compute $\hat{v} = W_j \cdot v$.
    2. Extend $\hat{v}$ by zeros so that $\hat{v}$ is of length $n$.
    3. Set $\hat{v} \leftarrow C_0 \hat{v}$ (see above).
    4. Truncate $\hat{v}$ to length $m$.
    5. Set $\hat{z} = \hat{z} + X_j \cdot \hat{v}$.
End for
6. Compute $\hat{z} = \hat{z} + \mathrm{diag}(c_j) v$.
```

7.3. Matrix-vector products with C. By relating C back to the original Toeplitz matrix, we note that matrix vector products with C can be computed as $Cv = STSv$. To multiply the matrix T with a vector, we could use the method of embedding T into a circulant matrix and using Fourier transforms. However, this requires that complex arithmetic be used to compute the product when T is real. Rather, we make use of the fast, real-arithmetic approach suggested in [14] for computing these products in $O(N \lg N)$ operations.

7.4. Variant of MINRES. In this subsection we present a variant of MINRES for solving the symmetrically preconditioned problem $M^{-1/2} C M^{-1/2} (M^{1/2} y) = M^{-1/2} z$ which involves matrix vector multiplies with M^{-1} rather than $M^{-1/2}$ (see [5, Section 10.3.1] and [7]). A variant of MR-II (see [8, 7]) for the symmetrically preconditioned problem involving only matrix vector multiplies with M^{-1} can be similarly derived.

```
Algorithm 5: Preconditioned MINRES
$y_0 = 0; r_0 = z; v_0 = M^{-1} r_0; d_0 = v_0$
$w_0 = C v_0; s_0 = w_0$
For $k = 0, \ldots,$ until convergence do
    $\alpha = \frac{v_k^T s_k}{w_k^T M^{-1} w_k}$
    $y_{k+1} = y_k + \alpha d_k$
    $r_{k+1} = r_k - \alpha w_k$
    $v_{k+1} = M^{-1} r_{k+1}$
    $s_{k+1} = C v_{k+1}$
    $\beta = \frac{v_{k+1}^T s_{k+1}}{v_k^T s_k}$
    $d_{k+1} = v_{k+1} + \beta d_k$
    $w_k = s_{k+1} + \beta w_k$
End for
```

8. Numerical Results. We compare the results of the preconditioned and unpreconditioned MINRES algorithm with the preconditioned and unpreconditioned CGLS algorithm. The numerical results were generated using Matlab and IEEE floating point double precision arithmetic. Since in our examples the exact solution to the noise-free problem was available, our measure of success in filtering noise is the relative error between the computed solution and the noise-free solution. In the experiments, we compare the results of MINRES with CGLS for Cauchy-like preconditioners of size m^* defined in this paper. The value of $m^* = 0$ corresponds to no preconditioning. In each example, we also give the results for the preconditioned method of Kilmer and O'Leary [16] and the method of Hanke, *et al* [9], for various values of m^*.

8.1. Example 1. For this example, we modified the matrix and exact solution of the signal processing example in [16] by dropping the last row and column of T and the last row of $\hat{f}$ and $\hat{g}$. The condition number of the new 255×255 matrix T is 4.4×10^5. We computed a noise vector e with Matlab's *randn* function and scaled it so that the noise level was 10^{-3}. We then computed $g = \hat{g} + e$.

Figure 5 is a sparsity plot of the magnitude of the entries of C. Note that not only is C nearly diagonally dominant, but pivoting need not be performed to permute the largest components of C to the leading principal submatrix.

The convergence of MINRES on the unpreconditioned system is indicated by the solid line in Fig 6. MINRES reaches its minimum relative error value of .232 at 41 iterations. The dashed line in the figure shows the convergence of CGLS on the unpreconditioned problem. After 119 iterations CGLS reaches its minimum relative error value of .223.

Table 1 compares the sensitivity of CGLS and MINRES to m^*. The results in the table illustrate that, for both methods, the number of iterations for the preconditioned system is substantially less than for the unpreconditioned system when m^* is chosen appropriately. Note that the preconditioned MINRES can yield a regularized solution with lower minimum relative error than unpreconditioned MINRES. The table also indicates that unpreconditioned CGLS can yield a slightly better, in terms of minimum relative error, regularized solution than MINRES, although it requires much more work to compute. Likewise, preconditioned CGLS, depending on m^*, can yield better regularized solutions the preconditioned MINRES in about the same number of

	MINRES		CGLS	
m^*	minimum rel. error	achieved at iter.	minimum rel. error.	achieved at iter.
0	.232	41	.223	119
40	.233	13	.224	24
47	.225	9	.224	13
54	.242	6	.224	7
61	.256	7	.221	7
68	.237	5	.228	8

TABLE 1

Convergence comparison of MINRES and CGLS for various values of m^* for Example 1.

	Method of [16]		Method of [9]	
m^*	minimum rel. error	achieved at iter.	minimum rel. error.	achieved at iter.
0	.223	119		
40	.224	75	.224	53
47	.224	58	.224	40
54	.224	82	.224	31
61	.238	105	.229	30
68	.289	13	.236	34

TABLE 2

Convergence comparison of preconditioned CGLS scheme of Kilmer and O'Leary and method of Hanke, et al *for various values of m^*, Example 1.*

iterations — however, each iteration of CGLS requires an extra matrix-vector product with C. The condition number of C_1 for $m^* = 47$ is about 87; the condition number of C_1 for $m^* = 61$ is about 2.5×10^4.

Table 2 gives the convergence results for the preconditioned CGLS scheme of [16] and for the preconditioned scheme of [9] for comparison purposes. Note that neither method does as well as the preconditioned MINRES or CGLS schemes mentioned in this paper in terms of reducing the error to a sufficient level within few enough iterations. Also, these methods are more expensive (by a constant factor) per iteration than preconditioned MINRES since they require an additional matrix-vector product with the Cauchy-like matrix or the Toeplitz matrix, respectively, and they compute using complex arithmetic. Further, the initialization cost of the preconditioner of [16] is higher. We note that since $n+1$ is a power of 2 and these latter 2 methods use FFT's, it would have been more efficient to augment T by a 1×1 identity and append a number to g, and solve the resulting system (see the footnote in the introduction of [16]).

8.2. Example 2. In this example, we used Hansen's Regularization Toolbox to generate a 512×512 symmetric Phillips Toeplitz matrix, and set T to be the 511×511 leading principal submatrix. The vector $\hat{f}$ was generated using Matlab's sin, cos and square functions in the following Matlab notation:

$$\hat{f} = (1 - \text{abs}(s)). * (1 + \cos(s * pi/3)) + \sin(s * pi/8). * (s + 3) + 9 * \text{square}(.4 * s.^2/50)$$

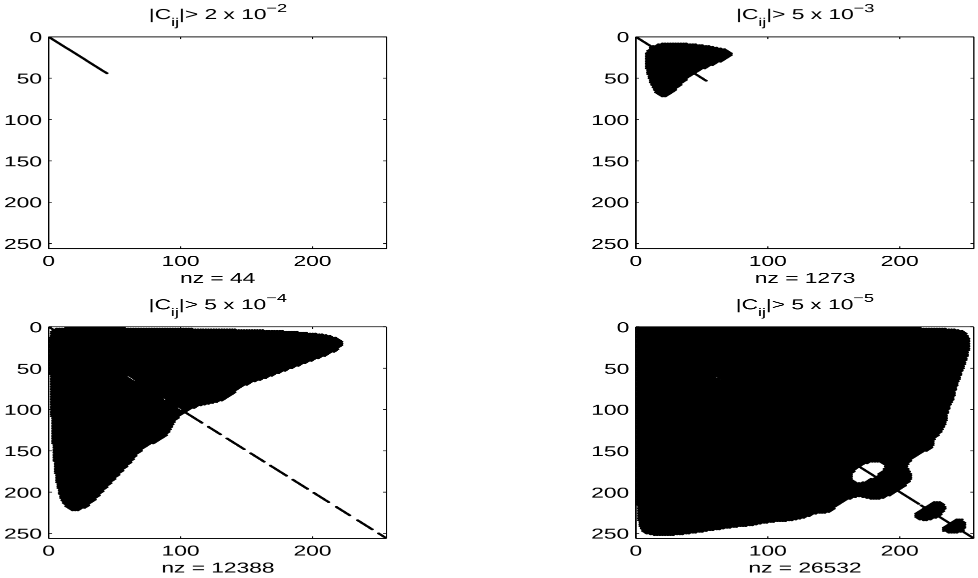

FIG. 5. *Spy plot of the magnitude of elements of* C*, Example 1.*

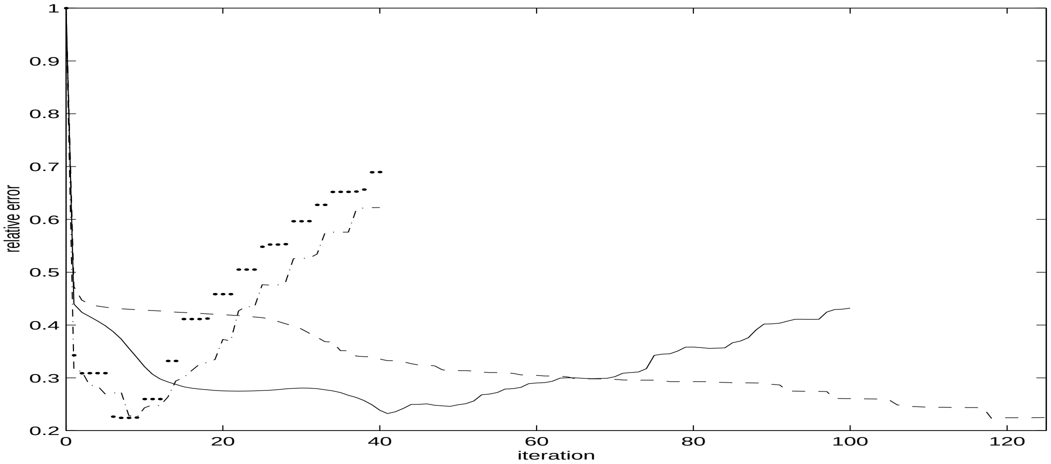

FIG. 6. *Convergence of MINRES (solid) and CGLS (dashed) for* $m^* = 0$*; preconditioned MINRES (dash-dot) with* $m^* = 47$*; and preconditioned CGLS (dotted) with* $m^* = 54$*.*

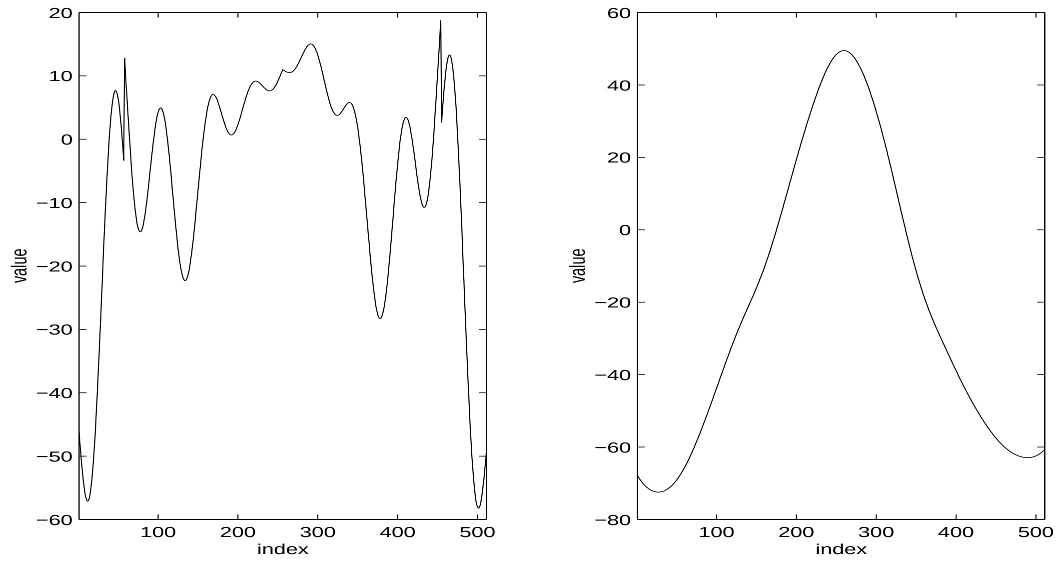

FIG. 7. *Uncontaminated data vector (right) and exact solution (left) for Example 2.*

where s was the vector of length 511 $s = [-25.5 : .1 : 25.5]$. The vectors $\hat{f}$ and $\hat{g} = T\hat{f}$ are is displayed in Figure 7. The noisy data g was formed by adding noise to the vector $\hat{g}$ where the noise level was 10^{-3}.

Figure 9 is a spy plot illustrating the magnitude of the elements in C. As in the previous example, no pivoting is needed to permute the largest magnitude entries into the leading principal submatrix of C.

Table 3 compares the minimum relative errors achieved for MINRES and CGLS with and without preconditioning. Note again that unpreconditioned CGLS achieves a lower minimum relative error than unpreconditioned MINRES. However, for several values of m^*, MINRES is able to reach a regularized solution with relative error less than unpreconditioned MINRES. With $m^* = 19$, preconditioned MINRES reaches a relative error of .162 after only 2 iterations, and it improves in 7 iterations to a minimum relative error of .088 (see Figure 7). On the other hand, for no value of m^* could preconditioned CGLS achieve a relative error of less than .107. In general, preconditioned CGLS required more iterations to achieve comparable regularized solutions, and at more work per iteration.

The results for the preconditioned scheme of [16] and for the method of [9] applied to Example 2 are shown in Table 4. The previous method can generate regularized solutions with smaller relative error than for unpreconditioned MINRES within 2 iterations (for example, if $m^* = 25$, the relative error is .149 after 2 iterations), but for no value of m^* do they achieve better minimum relative error values than preconditioned MINRES for $m^* = 19$. The method of Hanke *et al* is not very competitive with the other methods since it requires so many more iterations for each value of m^*.

Finally, Figure 10 illustrates how well our preconditioner clusters the eigenvalues and the singular values of the left preconditioned matrix.

9. Conclusions and Future Work. Preliminary results show that we have developed an efficient preconditioner for the regularized solution of discrete ill-posed

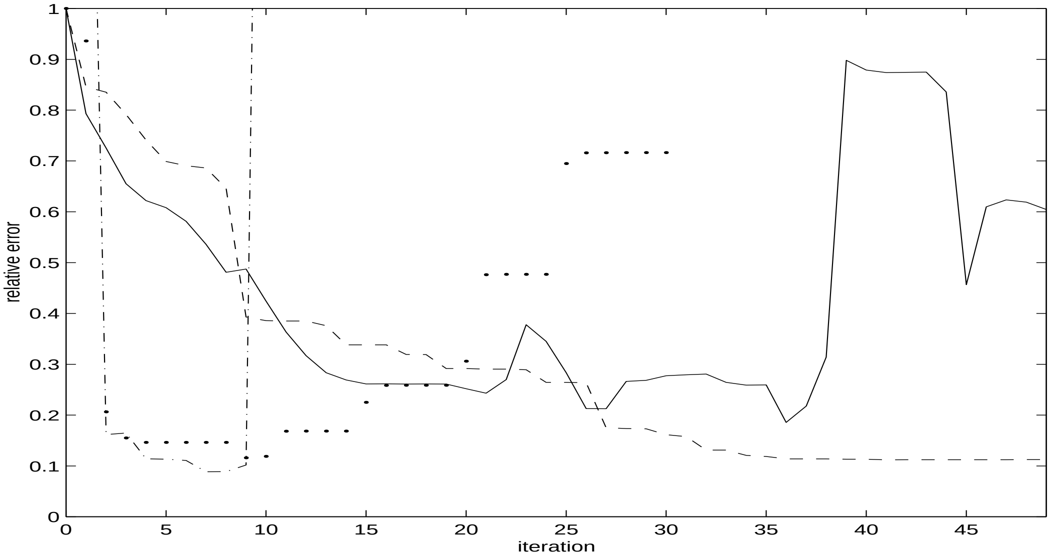

FIG. 8. *Convergence of MINRES (solid) and CGLS (dashed) for* $m^* = 0$*; preconditioned MINRES (dash-dot) with* $m^* = 19$*; preconditioned CGLS (dotted) with* $m^* = 31$.

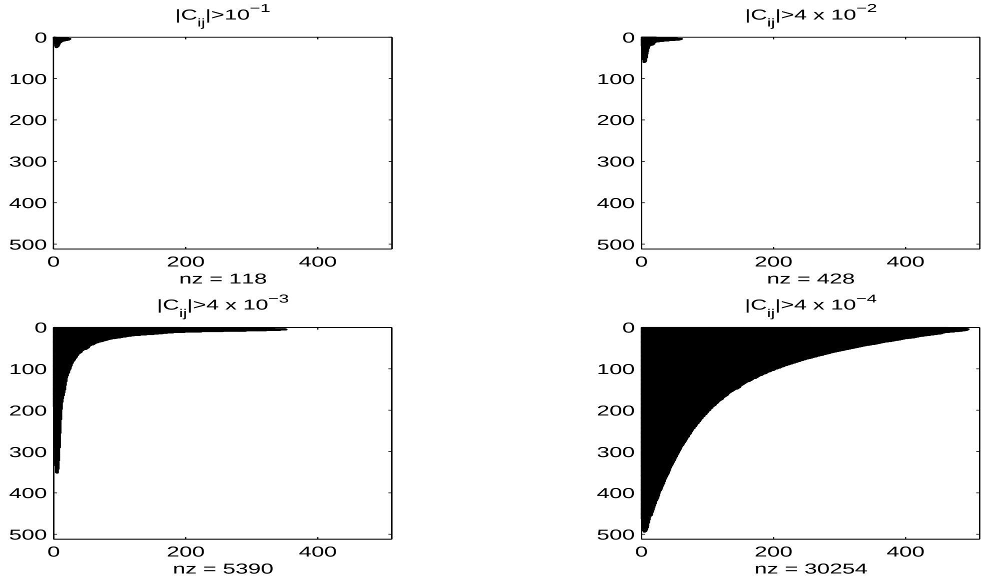

FIG. 9. *Spy plot of the magnitude of elements of* C*, Example 2.*

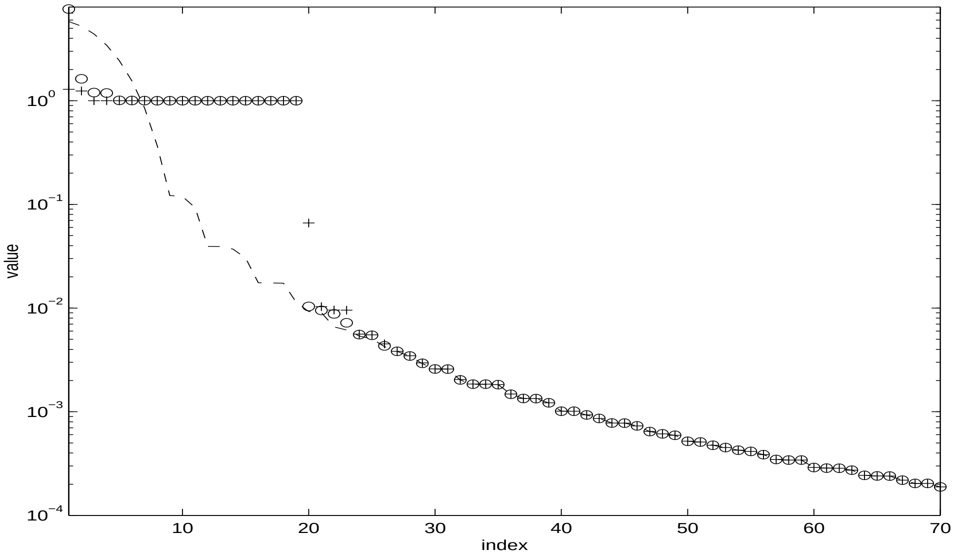

FIG. 10. *Largest (magnitude) 70 eigenvalues ('+') and singular values ('o') of the left preconditioned matrix for $m^* = 19$. Dotted line connects singular values (or absolute eigenvalues) of C.*

	MINRES		CGLS	
m^*	minimum rel. error	achieved at iter.	minimum rel. error.	achieved at iter.
0	.185	36	.112	41
16	.177	11	.111	21
19	.088	7	.107	17
22	.116	4	.117	14
25	.205	5	.121	14
28	.174	10	.116	11
31	.241	8	.116	9
34	.187	5	.164	13

TABLE 3
Convergence comparison of MINRES and CGLS for various values of m^*, Example 2.

	Method of [16]		Method of [9]	
m^*	minimum rel. error	achieved at iter.	minimum rel. error.	achieved at iter.
0	.112	41		
16	.110	19	.114	58
19	.109	15	.108	57
22	.115	15	.100	43
25	.123	12	.113	54
28	.136	9	.124	64
31	.136	4	.134	55
34	.159	8	.166	68

TABLE 4
Convergence comparison of preconditioned CGLS scheme of Kilmer and O'Leary and method of Hanke, et al *for various values of m^*, Example 2.*

problems involving symmetric Toeplitz matrices. We have introduced a preconditioned MINRES scheme to solve the symmetrically preconditioned problem. The theory and results predict that preconditioned MINRES can be an effective and efficient regularization scheme, with each iteration requiring fewer operations than preconditioned CGLS. In both examples, preconditioned MINRES for an appropriate value of m^* could achieve regularized solutions with smaller minimum relative errors than unpreconditioned MINRES.

We plan to generalize the results in this paper to the two-dimensional problems involving symmetric BTTB matrices.

Acknowledgment. The author would like to thank Vadim Olshevsky for suggesting the possibility of a symmetric analogue of the work in [16] and for providing reference to work in [14]. The author is also grateful to Dianne O'Leary for her helpful comments.

REFERENCES

[1] T. F. CHAN AND P. C. HANSEN, *A look-ahead Levinson algorithm for general Toeplitz systems*, IEEE Trans. Signal Processing, 40 (1992), pp. 1079–1090.

[2] T. FINCK, G. HEINIG, AND K. ROST, *An inversion formula and fast algorithms for Cauchy-Vandermonde matrices*, Linear Algebra Appl., 183 (1993), pp. 179–191.

[3] A. GERASOULIS, *A fast algorithm for generalized Hilbert matrices*, Math. Comput., 50 (1988), pp. 179–188.

[4] A. GERASOULIS, M. D. GRIGORIADIS, AND L. SUN, *A fast algorithm for Trummer's problem*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. s135–s137.

[5] G. H. GOLUB AND C. V. LOAN, *Matrix Computations*, Johns Hopkins University Press, 1989. Second Ed.

[6] M. H. GUTKNECHT AND M. HOCHBRUCK, *Look-ahead Levinson and Schur algorithms for non-Hermitian Toeplitz systems*, Numer. Math., 70 (1995), pp. 181–227.

[7] M. HANKE, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Longman Scientific and Technical, 1995.

[8] M. HANKE AND J. NAGY, *Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques*, Inverse Problems, 12 (1996), pp. 157–173.

[9] M. HANKE, J. NAGY, AND R. PLEMMONS, *Preconditioned iterative regularization for ill-posed problems*, Numerical Linear Algebra and Sci. Computing, (1993), pp. 141–163. L. Reichel, A. Ruttan, and R.S. Varga, editors.

[10] P. C. Hansen, *The discrete Picard condition for discrete ill-posed problems*, BIT, 30 (1990), pp. 658–672.

[11] P. C. Hansen and D. P. O'Leary, *The use of the l-curve in the regularization of discrete ill-posed problems*, SIAM J. Sci. Comput., 14 (1993), pp. 1487–1503.

[12] G. Heinig and A. Bojanczyk, *Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices. I. Transformations*, Linear Algebra Appl., 254 (1997), pp. 193–226.

[13] B. Jähne, *Digital Image Processing*, Springer-Verlag, 1991.

[14] T. Kailath and V. Olshevsky, *Displacement structure approach to discrete-trigonometric-transform based preconditioners of the G. Strang type and of T. Chan type*, (Sept. 1996). submitted, 1996.

[15] M. E. Kilmer, *Cauchy-like preconditioners for 2-dimensional ill-posed problems*, Tech. Report CS-TR-3776, University of Maryland, College Park, 1997.

[16] M. E. Kilmer and D. P. O'Leary, *Pivoted Cauchy-like preconditioners for regularized solution of ill-posed problems*, Tech. Report CS-TR-3682, University of Maryland, College Park, 1996.

[17] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[18] D. R. Sweet, *The use of pivoting to improve the numerical performance of Toeplitz matrix algorithms*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 468–493.

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

[20] J. M. Varah, *Pitfalls in the numerical solution of linear ill-posed problems*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 164–176.