PIVOTED CAUCHY-LIKE PRECONDITIONERS FOR REGULARIZED SOLUTION OF ILL-POSED PROBLEMS*

MISHA E. KILMER† AND DIANNE P. O'LEARY‡

Abstract. Many ill-posed problems are solved using a discretization that results in a least squares problem or a linear system involving a Toeplitz matrix. The exact solution to such problems is often hopelessly contaminated by noise, since the discretized problem is quite ill-conditioned, and noise components in the approximate null-space dominate the solution vector. Therefore we seek an approximate solution that does not have large components in these directions. We use a preconditioned conjugate gradient algorithm to compute such a regularized solution. An orthogonal change of coordinates transforms the Toeplitz matrix to a Cauchy-like matrix, and we choose our preconditioner to be a low rank Cauchy-like matrix determined in the course of Gu's fast modified complete pivoting algorithm. We show that if the kernel of the ill-posed problem is smooth, then this preconditioner has desirable properties: the largest singular values of the preconditioned matrix are clustered around one, the smallest singular values, corresponding to the noise subspace, remain small, and the signal and noise spaces are relatively unmixed. The preconditioned algorithm costs only $O(n \lg n)$ operations per iteration for a problem with n variables. The effectiveness of the preconditioner for filtering noise is demonstrated on three examples.

Key words. Regularization, ill-posed problems, Toeplitz, Cauchy-like, preconditioner, conjugate gradient, least squares

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

September 3, 1996

1. Introduction. In fields such as seismography, tomography, and signal processing, the process describing the acquisition of data can often be described by an integral equation of the first kind

$$\int_{\beta_{lo}}^{\beta_{up}} t(\alpha,\beta)\hat{f}(\beta)d\beta = \hat{g}(\alpha),$$

where t denotes the kernel, $\hat{f}$ the unknown input function, and $\hat{g}$ the output. When it is appropriately discretized, the equation becomes a system of n linear equations of the form

$$T\hat{f} = \hat{g}.$$

In applications, properties of the kernel and the discretization process often cause T to have a Toeplitz structure; that is, $t_{ij} = t_{i-j}$ for $1 \le i, j \le n$, and T is therefore constant along diagonals.

The discrete inverse problem is to recover $\hat{f}$, given $\hat{g}$ and T. However, the continuous problem is generally ill-posed: *i.e.* small changes in $\hat{g}$ cause arbitrarily large changes in $\hat{f}$. This is reflected in the discrete problem by ill-conditioning in the matrix T. The recovery of $\hat{f}$ then becomes a delicate matter since the recorded data will

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

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

‡ Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (oleary@cs.umd.edu).

likely have been contaminated by noise e. In this case, we have measured g rather than $\hat{g}$, where

$$T\hat{f} + e = \hat{g} + e = g. \tag{1}$$

Due to the ill-conditioning of T and the presence of noise, exact solution of the linear system will not lead to a reasonable approximation of $\hat{f}$. Rather, *regularization* is needed in order to compute an approximate solution f. Regularization can be thought of as exchanging the original, ill-posed problem for a more well-posed problem whose solution approximates the true solution. Many regularization methods, both direct and iterative, have been discussed in the literature; see, for example, [12, 15, 9, 5]. In this paper we will primarily be concerned with regularization via conjugate gradient iterations [7, 22, 29], where the regularization parameter is the number of iterations.

Toeplitz matrices have several properties convenient for iterative methods like conjugate gradients: multiplication of a Toeplitz matrix times a vector can be done in $O(n \lg n)$ operations, and circulant preconditioners can be quite efficient [25, 3]. There are some difficulties, though. The inverse of a Toeplitz matrix does not generally have Toeplitz structure, and the fast factorization algorithms for Toeplitz matrices can require as much as $O(n^3)$ flops if pivoting is used to improve stability; see [27, 11, 4], for example.

To overcome these difficulties, we make use of the fact that Toeplitz matrices are related to *Cauchy-like* matrices by fast orthogonal transformations [17, 8, 10]. Cauchy-like matrices, discussed in detail in §2, permit fast matrix-vector multiplication. But, in contrast to Toeplitz matrices, the inverse of a Cauchy-like matrix is Cauchy-like, and complete pivoting can be incorporated in its LDU factorization at a total cost of $O(n^2)$.

The focus of this paper is the development of a Cauchy-like preconditioner that can be used to accelerate convergence of the conjugate gradient iteration to a filtered approximate solution of a problem involving a Toeplitz matrix. The regularizing properties of conjugate gradients and our choice of preconditioner are discussed in §3. Each iteration of our algorithm takes $O(n \lg n)$ operations, and computational issues are discussed in §4. Section 5 contains numerical results and §6 presents conclusions and future work.

2. Transformation from Toeplitz to Cauchy-like structure. A Cauchy-like, or generalized Cauchy, matrix C has the form

$$C = \left(\frac{a_i^T b_j}{\omega_i - \theta_j} \right)_{1 \le i,j \le n} \quad (a_i, b_j \in \mathcal{C}^{\ell \times 1}; \omega_i, \theta_j \in \mathcal{C}). \tag{2}$$

It can also be defined as the unique solution of the *displacement equation*

$$\Omega C - C\Theta = AB^T \tag{3}$$

where

$$\Omega = \operatorname{diag}(\omega_1, \ldots, \omega_n), \Theta = \operatorname{diag}(\theta_1, \ldots, \theta_n), A = \begin{pmatrix} a_1^T \\ \vdots \\ a_n^T \end{pmatrix}, B = \begin{pmatrix} b_1^T \\ \vdots \\ b_n^T \end{pmatrix}.$$

The pair (A, B) is the *generator* of C with respect to Ω and Θ, and $\ell \le n$ is called the *displacement rank*. For the matrices and displacement equations of interest here, $\ell = 1$ or 2 [8].

We exploit three important properties of Cauchy-like matrices.

PROPERTY 1. *Row and column permutations of Cauchy-like matrices are Cauchy-like, as are leading principal submatrices.*

This property allows pivoting in fast algorithms for factoring Cauchy-like matrices [17, 8].

PROPERTY 2. *The inverse of a Cauchy-like matrix is Cauchy-like:*

$$C^{-1} = -\left(\frac{x_i^T w_j}{\theta_i - \omega_j}\right)_{1\le i,j\le n} \quad (x_i, w_j \in \mathcal{C}^{\ell\times 1}). \tag{4}$$

Heinig [17] gives an $O(n \lg^2 n)$ algorithm to compute X (with rows x_i^T) and W (with rows w_i^T) given A, B, Θ, and Ω, and explains how, using the FFT, a system involving a Cauchy-like matrix can be solved in $O(n \lg^2 n)$. However, the algorithm is very fragile. It can be unstable for large values of n and, even when used on a well conditioned matrix, may require pivoting to maintain stability [18, 1]. Alternatively, X and W can be determined from the relations

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

The third important property is that Toeplitz matrices also satisfy certain displacement equations [21, 8] which allow them to be transformed via fast Fourier transforms into Cauchy-like matrices [17, 8]:

PROPERTY 3. *Every Toeplitz matrix T satisfies an equation of the form*

$$R_1 T - T R_{-1} = AB^T \tag{6}$$

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

$$R_\delta = \begin{pmatrix} 0 & 0 & \dots & 0 & \delta \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}.$$

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

$$C = FTS_0^* F^*$$

that satisfies the displacement equation

$$S_1 C - C S_{-1} = (FA)(B^T S_0^* F^*)\,, \tag{7}$$

where

$$S_1 = \mathrm{diag}(1, e^{\frac{2\pi i}{n}}, \dots, e^{\frac{2\pi i}{n}(n-1)}),$$

$$S_{-1} = \mathrm{diag}(e^{\frac{\pi i}{n}}, \dots, e^{\frac{(2n-1)\pi i}{n}}),$$

$$S_0 = \mathrm{diag}(1, e^{\frac{\pi i}{n}}, \dots, e^{\frac{\pi i}{n}(n-1)}),$$

and F is the normalized inverse discrete Fourier transform matrix defined by

$$F = \frac{1}{\sqrt{n}}\left[\exp\left(\frac{2\pi i}{n}(j-1)(k-1)\right)\right]_{1\le j,k\le n}.$$

Gohberg, Kailath, and Olshevsky [8] suggest a stable $O(\ell n^2)$ partial pivoting algorithm to factor $C = PLU$. Sweet and Brent [26] show, however, that element growth in this algorithm depends not only on the magnitude of L and U, but on the generator for the Cauchy-like matrix. For our test matrices, partial pivoting alone did not provide the rank revealing information that we need.

Gu [10] presents an algorithm that can perform a fast $O(\ell n^2)$ variation of LU decomposition with complete pivoting. Recall that in complete pivoting, at every elimination step one chooses the largest element in the current submatrix as the pivot in order to reduce element growth. Gu proposes instead that one find an entry sufficiently large in magnitude by considering the largest 2-norm column of $\bar{A}\bar{B}^T$ corresponding to the part that remains to be factored at each step. This algorithm computes the factorization $C = PLUQ$ [10, Alg. 2] using only the readily determined generators (see §4), and Gu shows that it is efficient and numerically stable, provided that element growth in the computed factorization is not large. For our purposes it was convenient to set $D = diag(u_{11}, \ldots, u_{nn})$ and $U \leftarrow D^{-1}U$ to obtain the equivalent factorization $C = PLDUQ$.

3. Regularization and preconditioning. If we wanted to solve the linear system $Tf = g$ exactly, we would be finished: using the transformation to Cauchy-like form and the fast factorization algorithms described above, computing this solution would be an easy task. But the solution we seek is an approximate one, having noise filtering properties, so we choose to use an iterative method called CGLS which, in conjunction with an appropriate preconditioner, produces suitably filtered solutions.

Three assumptions will guide our analysis:

1. The matrix T has been normalized so that its largest singular value 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 like the singular values [30, 14].
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.

We need to define the *signal* and *noise* subspaces. Using (1), let $T = \bar{U}\Sigma\bar{V}^T$ be the singular value decomposition of T, and expand the data and the noise in the basis created by the columns of $\bar{V}$:

$$\hat{g} = \sum_{i=1}^{n} \hat{\gamma}_i v_i, \quad e = \sum_{i=1}^{n} \eta_i v_i,$$

with $\hat{\gamma} = \bar{V}^T\hat{g}$ and $\eta = \bar{V}^T e$. Under the white noise assumption, the coefficients η_i are roughly constant in size, while the discrete Picard condition tells us that the $\hat{\gamma}_i$ go to zero at least as fast as the singular values σ_i. Thus, components for which $\hat{\gamma}_i$ is of the same order as η_i are obscured by noise. Let m be such that $|\hat{\gamma}_i| \gg |\eta_i|$ for $i = 1 \ldots m$ and $|\hat{\gamma}_i| \approx |\eta_i|$ or $|\hat{\gamma}_i| < |\eta_i|$ for $i = \bar{m}+1 \ldots n$. Then we say that the last $n - \bar{m}$ columns of $\bar{V}$ span the *noise subspace*, while the other columns span the *signal subspace*. The basis for the signal subspace is further partitioned into the first m columns and the remaining $\bar{m} - m$, which correspond to a *transition subspace* that is generally difficult to resolve unless there is a gap in the singular value spectrum.

3.1. Regularization by preconditioned conjugate gradients. The standard conjugate gradient (CG) method [19] is an iterative method for solving systems of linear equations for which the matrix is symmetric positive definite. If the matrix is not symmetric positive definite, one can use a variant of standard CG which solves the normal equations in factored form. We refer to the resulting algorithm as CGLS [19]. If the discrete Picard condition holds, then CGLS acts as an iterative regularization method with the iteration index taking the role of the regularization parameter [7, 13, 15]. Convergence is governed by the spread and clustering of the singular values [28]. Therefore, preconditioning is often applied in an effort to cluster the singular values, thus speeding convergence.

In the context of an ill-conditioned matrix T, we require a preconditioner for CGLS which clusters the largest m singular values while leaving the small singular values, and with them, the noise subspace, relatively unchanged. In this case, the first few iterations of CGLS will quickly capture the solution lying within the subspace spanned by the first m columns of V. A modest number of subsequent iterations will provide improvement over the transition subspace, without significant contamination from the noise subspace.

3.2. The preconditioner. Given the Toeplitz matrix T, let $\tilde{C} = FTS_0^* F^*$ be its corresponding Cauchy-like matrix. Solving $Tf = g$ is then equivalent to solving

$$\tilde{C} F S_0 f = Fg.$$

Note that since F and S_0 are unitary matrices, then

$$\tilde{C} = (F\bar{U})\Sigma \bar{V}^T (S_0^* F^*);$$

that is, $T = \bar{U}\Sigma\bar{V}^T$ and $\tilde{C}$ have the same singular values, and there is no mixing of signal and noise subspaces.

A factorization of $\tilde{C}$ using a modified complete pivoting strategy may lead to an interchange of rows (specified by a permutation matrix P) and columns (specified by a permutation matrix Q). Setting $C = P^T \tilde{C} Q^T$, $y = QFS_0 f$, and $z = P^T Fg$, the problem we wish to solve is

$$Cy = z. \tag{8}$$

We choose a preconditioner M for the left so that

$$M^{-1} C y = M^{-1} z$$

and apply CGLS to the corresponding normal equations

$$(M^{-1}C)^* (M^{-1}C) y = (M^{-1}C)^* M^{-1} z. \tag{9}$$

Our choice of preconditioner M is derived from the leading $m \times m$ submatrix of Gu's modified complete pivoting LDU factorization of the matrix C as follows. Let $C = LDU$ and write this equation in block form, where the upper left blocks are $m \times m$:

$$\begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ L_2 & L_3 \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} U_1 & U_2 \\ 0 & U_3 \end{bmatrix}. \tag{10}$$

Here L_1, L_3 are lower triangular, U_1, U_3 are upper triangular, and D_1 and D_2 are diagonal. We choose as our preconditioner the matrix

$$M = \begin{bmatrix} L_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} U_1 & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ 0 & I \end{bmatrix}.$$

3.3. Properties of the preconditioner. We begin with some theorems about the clustering of the singular values of $M^{-1}C$. It is useful to decompose the matrix $(M^{-1}C)^*(M^{-1}C)$ into the matrix sum

$$\left[\begin{array}{cc} I & C_1^{-1}C_2 \\ (C_1^{-1}C_2)^* & (C_1^{-1}C_2)^*(C_1^{-1}C_2) \end{array}\right] + \left[\begin{array}{cc} C_3^*C_3 & C_3^*C_4 \\ C_4^*C_3 & C_4^*C_4 \end{array}\right] \equiv E_1 + E_2 \tag{11}$$

using the block partitioning of the previous section.

Let ϵ_i be the sum of the absolute values of the entries in row i of $C_1^{-1}C_2$, let ϵ_{max} be the largest of these quantities, and let $\hat{s}$ be the largest such row sum for E_2. The case of interest to us is when these quantities are reasonably small.

We denote the k-th largest singular value of a matrix Z by $\sigma_k(Z)$, and the k-th largest eigenvalue by $\lambda_k(Z)$.

THEOREM 3.1. *If $\|C_1^{-1}C_2\|_2 < 1$ then the m largest singular values of $M^{-1}C$ lie in the interval $\left[1, \sqrt{1+\epsilon_{max}+\hat{s}}\right]$.*

Proof: The upper bound can be obtained by applying Gershgorin's theorem [24][IV.2.1] to bound the eigenvalues of the matrix $(M^{-1}C)^*(M^{-1}C)$, and then taking square roots. The lower bound is somewhat more interesting.

The matrices E_1 and E_2 are Hermitian positive semidefinite, and from the representations

$$E_1 = \left[\begin{array}{cc} I & 0 \\ (C_1^{-1}C_2)^* & 0 \end{array}\right]\left[\begin{array}{cc} I & (C_1^{-1}C_2) \\ 0 & 0 \end{array}\right] \text{ and } E_2 = \left[\begin{array}{cc} C_3^* & 0 \\ C_4^* & 0 \end{array}\right]\left[\begin{array}{cc} C_3 & C_4 \\ 0 & 0 \end{array}\right],$$

it is clear that they have rank at most m and $n-m$, respectively.

By Corollary IV.4.9 [24], we know that

$$\lambda_i(E_1) \le \lambda_i((M^{-1}C)^*(M^{-1}C)) \tag{12}$$

We need to show that $\lambda_k(E_1) \ge 1$. If Y_1 and Y_2 are two $n \times n$ matrices and the rank of Y_2 is $n-m$ then a theorem of Weyl [20, Thm. 3.3.16] implies $\sigma_n(Y_1+Y_2) \le \sigma_m(Y_1)$. Now set

$$Y_1 = \left[\begin{array}{cc} I & C_1^{-1}C_2 \\ 0 & 0 \end{array}\right], Y_2 = \left[\begin{array}{cc} 0 & -C_1^{-1}C_2 \\ 0 & I \end{array}\right],$$

and notice that the eigenvalues of E_1 are the squares of the singular values of Y_1. But Y_1+Y_2 is the $n \times n$ identity matrix, so by Weyl's result we obtain $\sigma_m(Y_1) \ge 1$. Thus, $\lambda_i(E_1) \ge 1$ for $i = 1, \dots, m$ and our conclusion follows from (12). $\Box$

We now study the extent to which preconditioning by M mixes the signal and noise subspaces.

THEOREM 3.2. *Let k be the dimension of the noise subspace and let*

$$\begin{aligned} C &= [Q_1\, Q_2\, Q_3] \left[\begin{array}{ccc} \Sigma_1 & 0 & 0 \\ 0 & \Sigma_2 & 0 \\ 0 & 0 & \Sigma_3 \end{array}\right]\left[\begin{array}{c} V_1^* \\ V_2^* \\ V_3^* \end{array}\right], \\ M^{-1}C &= \left[\hat{Q}_1\, \hat{Q}_2\, \hat{Q}_3\right] \left[\begin{array}{ccc} \hat{\Sigma}_1 & 0 & 0 \\ 0 & \hat{\Sigma}_2 & 0 \\ 0 & 0 & \hat{\Sigma}_3 \end{array}\right]\left[\begin{array}{c} \hat{V}_1^* \\ \hat{V}_2^* \\ \hat{V}_3^* \end{array}\right] \end{aligned}$$

be singular value decompositions with $V_3, \hat{V}_3 \in \mathcal{C}^{n \times k}$ and $V_1, \hat{V}_1 \in \mathcal{C}^{n \times m}$. Then

$$\|V_1^*\hat{V}_3\|_2 \le \frac{\hat{\sigma}_{n-k+1}}{\sigma_m} \max\{1, \|C_1\|_2\}.$$

Proof: Using the decompositions we have

$$\begin{aligned} V_1^* \hat{V}_3 &= (V_1^* C^{-1}) M (M^{-1} C \hat{V}_3) \\ &= \Sigma_1^{-1} Q_1^* M \hat{Q}_3 \hat{\Sigma}_3. \end{aligned}$$

Since Q_1 and $\hat{Q}_3$ have orthonormal columns, it follows that

$$\|V_1 \hat{V}_3\|_2 \leq \frac{\hat{\sigma}_{n-k+1}}{\sigma_m} \|M\|_2 = \frac{\hat{\sigma}_{n-k+1}}{\sigma_m} (max\{1, \|C_1\|_2). \ \Box$$

Next we show that $\hat{\sigma}_j \approx \sigma_j$ for σ_j corresponding to the noise subspace, and thus $\hat{\sigma}_{n-k+1}$ is small. Thus, if C_1 is well-conditioned, then we are guaranteed that the signal and noise subspaces remain unmixed.

THEOREM 3.3. *The $(m+i)$th singular value of each of the matrices C and $M^{-1}C$ lies in the interval $[0, \sigma_i(E_2)]$, for $i = 1, \ldots, n-m$.*

Proof: Two theorems due to Weyl for Hermitian matrices Z, Y_1, and Y_2 with $Z = Y_1 + Y_2$ say

$$\lambda_{k+j-1}(Z) \leq \lambda_k(Y_1) + \lambda_j(Y_2) \quad \text{[24, p.210] and}$$

$$\lambda_n(Y_2) + \lambda_k(Y_1) \leq \lambda_k(Z) \quad \text{[24, Cor. IV.4.9]} \ .$$

Now from the decomposition in Equation (11), we see $\lambda_n(E_2) = 0$ and $\lambda_{m+1}(E_1) = 0$, and thus

$$0 \leq \lambda_{m+i}((M^{-1}C)^*(M^{-1}C)) \leq \lambda_{m+1}(E_1) + \lambda_i(E_2) = \lambda_i(E_2)$$

for $i = 1, \ldots, (n-m)$.

Also,

$$C^* C = \begin{bmatrix} C_1^* & 0 \\ C_2^* & 0 \end{bmatrix} \begin{bmatrix} C_1 & C_2 \\ 0 & 0 \end{bmatrix} + E_2.$$

We therefore likewise obtain

$$0 \leq \lambda_{m+i}(C^* C) \leq \lambda_i(E_2).$$

The proof is completed by taking square roots. $\Box$

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_2 are small. We now discuss to what extent these conditions hold for integral equation discretizations.

PROPERTY 4. *Let $\tilde{C}$ be a Cauchy-like matrix corresponding to a real Toeplitz matrix T that results from discretization of a smooth kernel t, normalized so that the maximum element of T is one. Then for n sufficiently large, there exists $\epsilon \ll 1$ and $m \ll n$ such that all elements of $\tilde{C}$ are less than ϵ in magnitude except for those located in four corner blocks of total dimension $m \times m$.*

To understand why this is true, recall that if $\tilde{A}$ and $\tilde{B}$ are the generators of $\tilde{C}$, where $\tilde{G} = \tilde{A}\tilde{B}^T$, the magnitude of the (k,j)-entry of $\tilde{C}$ is $|\tilde{C}_{kj}| = \frac{|\tilde{a}_k^T \tilde{b}_j|}{|\omega_k - \theta_j|}$. Thus the largest entries in $\tilde{C}$ appear where the numerator is large or the denominator is small.

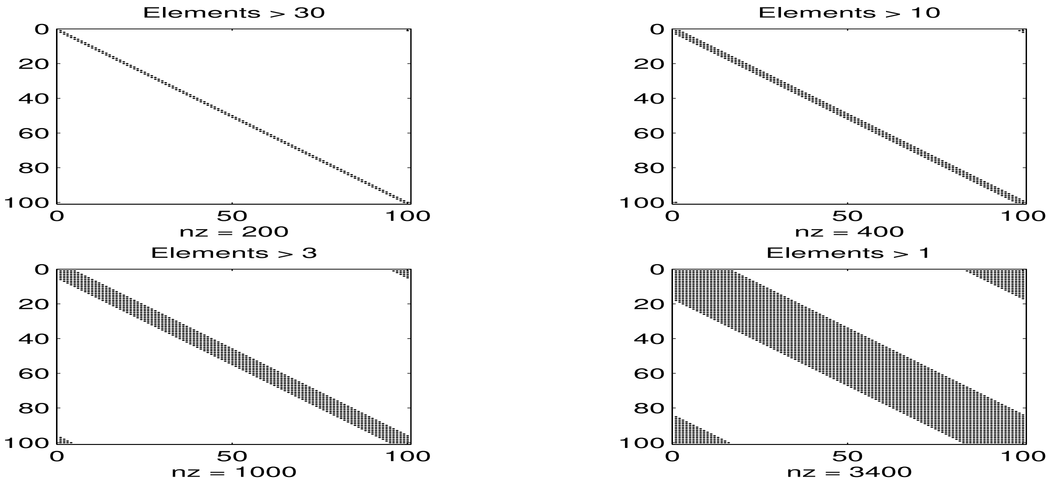

FIG. 1. *Plot revealing* $\left[\frac{1}{|\omega_k-\theta_j|}\right]_{k,j=1,\ldots,n} > tol, n = 100$ *for tol=30,10,3,and 1.*

The denominator of $\tilde{C}_{kj}$ is $|\omega_k - \theta_j| = |1 - e^{\frac{\pi i}{n}(2(j-k)+1)}|$, which is bounded above by 2. Its smallest entries are attained for $|k-j| \approx 0$ or n, but there are very few small values. In fact, direct computation shows that for $n \geq 100$, at least 95% of the entries in the first row have denominators in the range $[10^{-1}, 2]$, and the other rows have even more in this range. Figure 1 plots values of the matrix $\left[\frac{1}{|\omega_k-\theta_j|}\right]_{k,j=1,\ldots,n}$ for $n = 100$ above given tolerance levels. As expected, there are very few large values, and these occur only near the diagonal and the corners of the matrix.

Now consider the numerators. The formulas for A and B are determined from direct computation in (6). The first column of A is the first unit vector, and the second column is given by

$$(13) \qquad [0, t_{n-1}, t_{n-2}, \ldots, t_{p-1}, \ldots, t_1]^T + [t_0, t_{-1}, t_{-2}, \ldots, t_{-(q-1)}, \ldots, t_{-(n-1)}]^T$$

The first column of B is

$$(14) \qquad [t_{-(n-1)}, t_{-(n-2)}, \ldots, t_{-(q-1)}, \ldots, t_{-1}, t_0] - [t_1, t_2, \ldots, t_{p-1}, \ldots, t_{n-1}, 0]$$

and the second column is the last unit vector. The generators for $\tilde{C}$ are then $\tilde{A} \equiv FA$ and $\tilde{B} \equiv conj(FS_0)B$, where $conj(\cdot)$ denotes complex conjugation, with F and S_0 as described in Property 3. Therefore, the numerators are

$$|\tilde{a}_k^T \tilde{b}_j| = |\frac{1}{\sqrt{n}} conj(\zeta)_j + \frac{1}{\sqrt{n}} e^{\frac{(n-1)}{n}\pi i(1-2j)} \nu_k|,$$

where ν_k is the k^{th} entry in the second column of $\tilde{A}$ and ζ_j is the j^{th} entry in the first column of FS_0B. Thus it is the normalized inverse Fourier coefficients of the second column of A and first column of S_0B which determine the magnitude of the numerators, and if t is smooth, these will be large only for small indices j and k.

Therefore,

$$\frac{|\tilde{a}_k^T \tilde{b}_j|}{|\omega_k - \theta_j|} \leq \frac{1}{\sqrt{n}}(|\nu_k| + |\zeta_j|) \ll 1$$

away from the corners. Thus $\tilde{C}$ can be permuted to contain the large elements in the upper left block, and any pivoting strategy that produces such a permutation will give a suitable preconditioner for our scheme.

We have observed that if Gu's algorithm is applied to a matrix with this structure, then C_1 will contain the four corner blocks. The interested reader is referred to [10] for details on the complete pivoting strategy, but the key fact is that Gu makes his pivoting decisions based on the size of elements in the generator $\bar{A}\bar{B}^T$ corresponding to the block that remains to be factored. The resulting Cauchy-like preconditioner C_1 for the matrix C then has the properties that the first m singular values of the preconditioned matrix are clustered, and that the invariant subspace corresponding to small singular values of C is not much perturbed. Thus we expect that the initial iterations of CGLS will produce a solution that is a good approximation to the noise-free solution.

4. Algorithmic issues. Our algorithm is as follows:

Algorithm 1: Solving $Tf = g$	
1.	Compute the generators for the matrix $\tilde{C} = FTS_0^*F^*$ using (13) and (14).
2.	Determine an index m to define the size of the partial factorization of $\tilde{C}$ and factor $\tilde{C} = PLDUQ$.
3.	Set $C = P^T\tilde{C}Q^T$ and $z = P^TFg$.
4.	Determine the $m \times m$ leading principal submatrix, C_1, of C and let $M = \begin{bmatrix} C_1 & 0 \\ 0 & I \end{bmatrix}$. (See (10).)
5.	Compute an approximate solution $\tilde{y}$ to $M^{-1}Cy = M^{-1}z$ using a few steps of CGLS.
6.	The approximate solution in the original coordinate system is $f = S_0^*F^*Q^T\tilde{y}$.

When to stop the CGLS iteration in order to get the best approximate solution is a well-studied but open question (for instance, see [16] and the references therein). We do not solve this problem, but we consider the other algorithmic issues in the following subsections.

4.1. Determining the size of C_1. The choice of the parameter m determines the number of clustered singular values in the preconditioned system. It influences the amount of work per iteration, but perhaps more importantly, the mixing of signal and noise subspaces. We use a simple heuristic in our numerical experiments. We compute the Fourier Transform of the data vector $\hat{g}$ and determine the index m for which the Fourier coefficients start to level off. This is presumed to be the noise level, and the factorization is truncated here.

4.2. Computing the preconditioner. Since $\tilde{C}$ satisfies the displacement equation (3), with $\Omega = S_1$ and $\Theta = S_{-1}$, it follows that C_1 satisfies

$$\Omega_1 C_1 - C_1 \Theta_1 = A_1 B_1^T,$$

where Ω_1 and Θ_1 are the leading principal submatrices of $P^T\Omega P$ and $Q\Theta Q^T$ respectively, and A_1 and B_1 contain the first m rows of $P^T\tilde{A}$ and $Q^T\tilde{B}$ respectively.

Thus the matrix C_1^{-1} has entries

$$C_1^{-1} = -\left(\frac{x_i^T w_j}{\tilde{\theta}_i - \tilde{\omega}_j}\right)_{1 \le i,j \le n}, \tag{15}$$

where $\tilde{\theta}_i$ and $\tilde{\omega}_j$ are the elements of Θ and Ω that appear in Θ_1 and Ω_1 respectively and, from (5), the vectors x_i^T and w_j^T are rows of X_1 and W_1 defined as

$$C_1 X_1 = A_1, \;\; W_1^T C_1 = B_1^T. \tag{16}$$

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.

4.3. Applying the preconditioner. Let r be a vector of length m and assume that no pivoting was done when $\tilde{C}$ was factored. Heinig [17] states that $C_1^{-1}r$ may be written as

$$C_1^{-1}r = \sum_{j=1}^{\ell} -(X_1)_j \cdot (C_0 (W_1)_j \cdot r)$$

where $(X_1)_j$ is the jth column of X_1, $(W_1)_j$ is the jth column of W_1, and C_0 is the Cauchy matrix $C_0 = \left(\frac{1}{\theta_i - \omega_j}\right)_{1 \le i,j \le m}$. The notation $\cdot$ denotes the componentwise product of two vectors.

Fast multiplication by the matrix C_0 requires finding the coefficients of a polynomial whose roots are the elements of Θ_1 and Ω_1 [6], and this process can be unstable. To avoid this difficulty, realizing that the elements of S_{-1} and S_1 are roots of unity, we extend C_0 to a matrix of size $n \times n$ satisfying the displacement equation (2) with $\Omega = S_{-1}$ and $\Theta = S_1$, and we develop a mathematically equivalent algorithm for computing $s = C_1^{-1}r$:

Algorithm 2: Forming $s = C_1^{-1}r$
Set $s = 0$.
For $j = 1, \ldots, \ell$, do
1. Compute $\hat{r} = W_j \cdot r$.
2. Extend $\hat{r}$ by zeros so that $\hat{r}$ is of length n.
3. Set $\hat{r} \leftarrow C_0 \hat{r}$.
4. Truncate $\hat{r}$ to length m.
5. Set $s = s + X_j \cdot \hat{r}$.

The product $C_1^{-*}r$ can be computed similarly.

If pivoting was done during factorization, the vector $\hat{r}$ should be multiplied by Q after Step 2 and by P after Step 4.

This formulation allows $C_1^{-1}r$ to be computed in $O(n \lg n)$ operations in a stable manner, using an observation of Finck, Heinig, and Rost [6] that any Cauchy-like matrix can be factored as

$$C_0 = \mathrm{diag}(h(\theta_1), \ldots, h(\theta_n))^{-1} V(\theta) H V(\omega)^T, \tag{17}$$

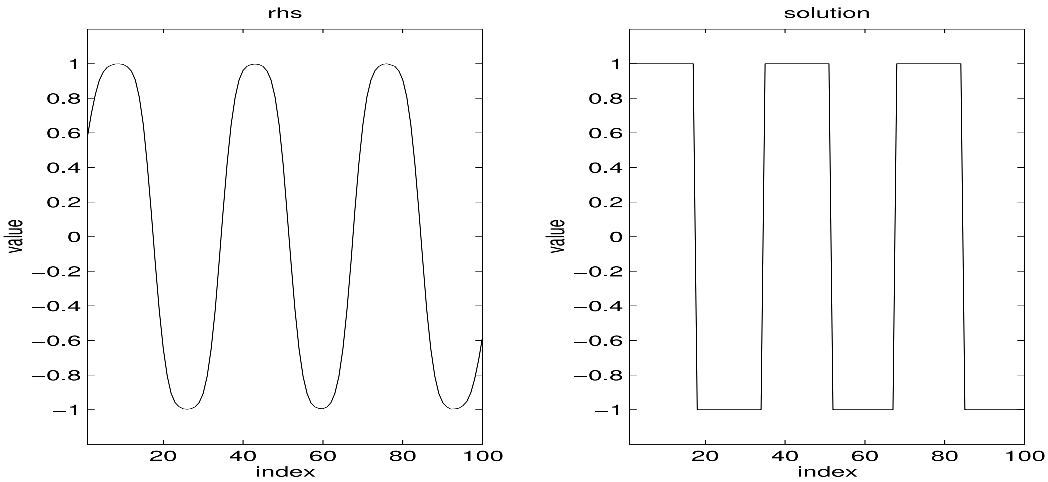

FIG. 2. *Uncontaminated data vector (left) and exact solution vector (right) for Example 1.*

m	minimum rel. error	achieved at iter.
0	2.13×10^{-1}	76
27	2.13×10^{-1}	31
39	2.17×10^{-1}	9
42	2.12×10^{-1}	6
43	2.02×10^{-1}	7
44	2.77×10^{-1}	8
45	2.76×10^{-1}	5

TABLE 1
Minimum relative errors achieved for various values of m, Example 1.

where $V(\omega)$ and $V(\theta)$ are the Vandermonde matrices whose second columns contain the diagonal elements of Ω and Θ, respectively. The matrix H is a Hankel matrix, i.e., one in which elements on the antidiagonals are constant. The first row is equal to the coefficients of the polynomial $h(u) = \prod_{i=1}^{n}(u - \omega_i)$ except for the leading one. Since, from Property 3, Ω and Θ contain roots of unity, products of the matrix C_0 with a vector are very simple to compute:

- $h(u) = u^n - 1$, so H has a single non-zero diagonal extending south-west to north-east.
- $V(\omega)^T$ is F, the normalized, discrete, inverse Fourier Transform matrix.
- $V(\theta)$ is the matrix product FS_0, where the diagonal matrix S_0 is defined in Property 3.

Thus products $C_0\hat{r}$ can be computed stably in $O(n \lg n)$ operations. Since $\ell = 2$ at most, the preconditioner can be applied to a vector in $O(n \lg n)$ operations, provided one knows X and W. This is the same order as the number of operations to apply C to a vector, since $C = PFTS_0^* F^* Q$ and the product of a Toeplitz matrix with a vector can be computed in $O(n \lg n)$ operations by embedding the matrix in a circulant matrix [2]. Thus, each iteration of CGLS costs $O(n \lg n)$ operations.

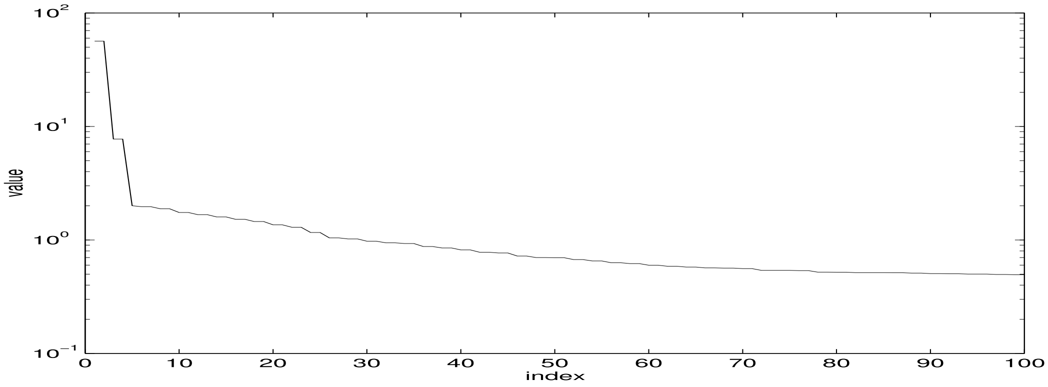

FIG. 3. *Fourier coefficients of the noisy data for Example 1.*

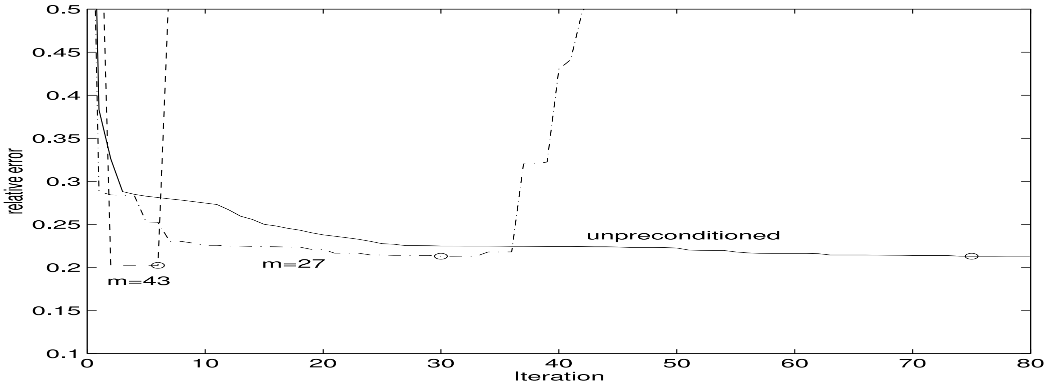

FIG. 4. *Relative error in computed solution for $m = 0$, $m = 27$, and $m = 43$, Example 1.*

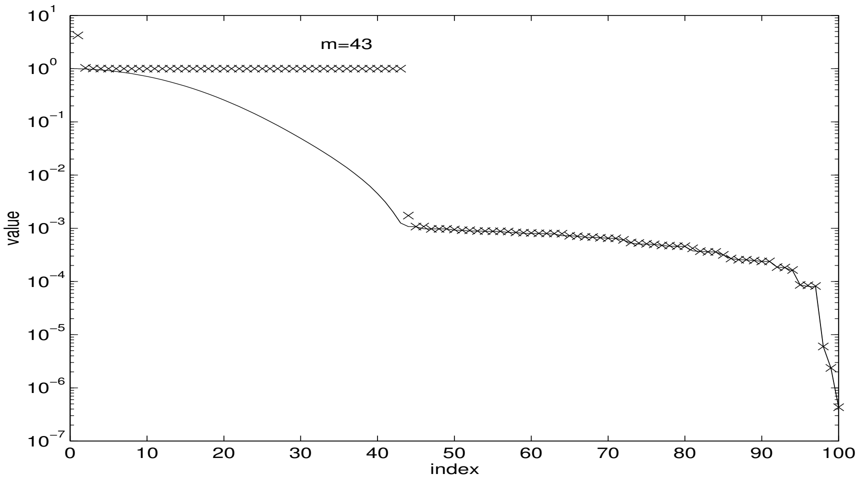

FIG. 5. *Singular values of C (solid line) and $M^{-1}C$ ($\times$'s) for Example 1, $m = 43$.*

5. Numerical results. In this section we summarize results of our algorithm on three test problems using Matlab and IEEE floating point double precision arithmetic. Our measure of success in filtering noise is the *relative error*, the 2-norm of the difference between the computed estimate f and the vector $\hat{f}$ corresponding to zero noise, divided by the 2-norm of $\hat{f}$. In each case, we apply the CGLS iteration with Cauchy-like preconditioner of size m. The value $m = 0$ corresponds to no preconditioning.

5.1. Signal processing example. As mentioned in the introduction, Toeplitz matrices often arise in the signal processing (1-dimensional image reconstruction problems). As an example, we consider the 100×100 Toeplitz matrix T whose entries are defined by

$$t_{i,j} = \begin{cases} \frac{4}{51}\phi(\alpha_i - \beta_j) & \text{if } |i-j| \leq 8, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\alpha_i = \beta_i = \frac{4i}{51}, \; i = 1, 2, \ldots, 100,$$

and

$$\phi(\gamma) = \frac{1}{2\sqrt{\pi}\delta}\exp(-\frac{\gamma^2}{4\delta^2}), \quad \delta = 0.15\,.$$

This matrix is the one used in Example 4 of [2]. The authors note that such matrices may occur in image restoration contexts as "prototype problems" and are used to model certain degradations in the recorded image.

The condition number of T is approximately 2.4×10^6. We wish to solve the equation $Tf = g$ where g denotes the noisy data vector for which $\|e\|_2/\|\hat{g}\|_2$, the noise level, is about 10^{-3}. The uncorrupted data, $\hat{g}$, and exact numerical solution, $\hat{f}$, are displayed in Figure 2[1]. The Fourier coefficients of g are shown in Figure 3. Using these coefficients, an appropriate cutoff value m was determined as explained in §4.1.

The solid line in Figure 4 shows the convergence of CGLS on the unpreconditioned Toeplitz system, where the ring on the line indicates the iteration at which the minimal value of the relative error, 2.13×10^{-1}, was achieved. Convergence of CGLS on the preconditioned system involving the Cauchy-like matrix is also shown in Figure 4 for two different values of m. Table 1 gives an idea of the sensitivity of the algorithm to the choice of m, with $m = 43$ being optimal in the sense of achieving minimal relative error among all choices of preconditioner. The number of iterations for the preconditioned system is substantially less than for the unpreconditioned.

The singular values of T and of the preconditioned matrix $M^{-1}C$ for $m = 43$ are shown in Figure 5. As predicted by the theory in Section 3.3, the first 43 singular values of $M^{-1}C$ are clustered very tightly around one and the smallest singular values have been left virtually untouched.

5.2. Phillips test problem. Next we consider the discretized version of the well-known first-kind Fredholm integral equation studied by D.L. Phillips [23]. The

[1] We first determined $\hat{f}$ using Matlab's square function, $\hat{f} = square(2\pi v * .3)$ with $v = [0:.1:9.9]$, then computed $\hat{g} = T\hat{f}$.

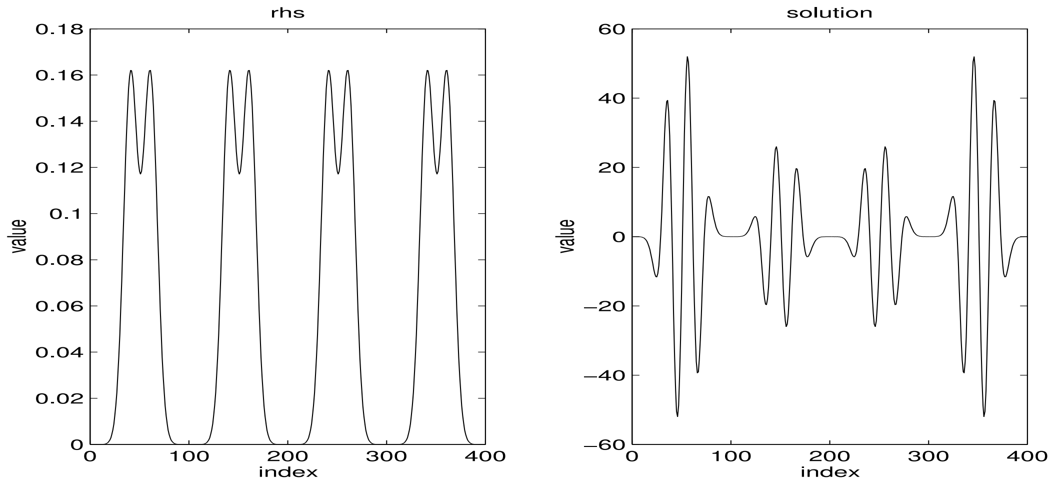

FIG. 6. *Uncontaminated data vector (left) and exact solution vector (right) for Example 2.*

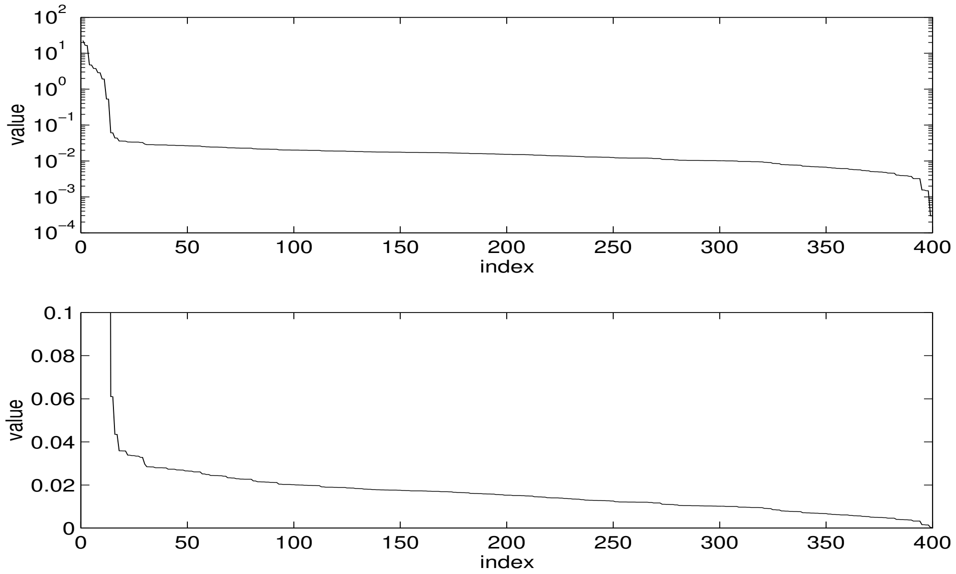

FIG. 7. *Fourier coefficients of the noisy data for Example 2, two different scales.*

m	minimum rel. error	achieved at iter.
0	5.71×10^{-2}	293
34	5.64×10^{-2}	53
48	4.68×10^{-2}	26
52	4.21×10^{-2}	16
56	3.01×10^{-2}	11
58	3.05×10^{-2}	9
68	5.53×10^{-2}	10

TABLE 2
Minimum relative errors achieved for various values of m, Example 2.

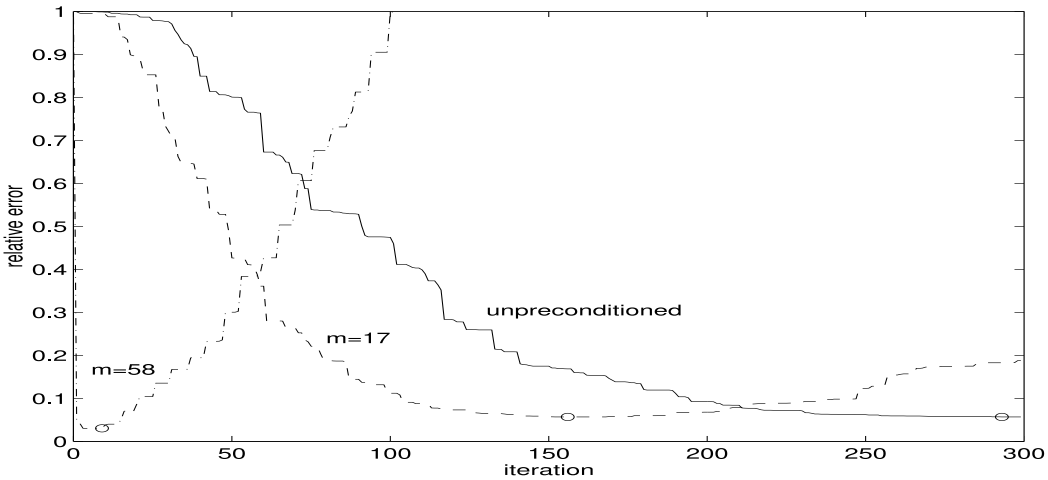

FIG. 8. *Relative error in computed solution for $m = 0$, $m = 17$, and $m = 58$, Example 2.*

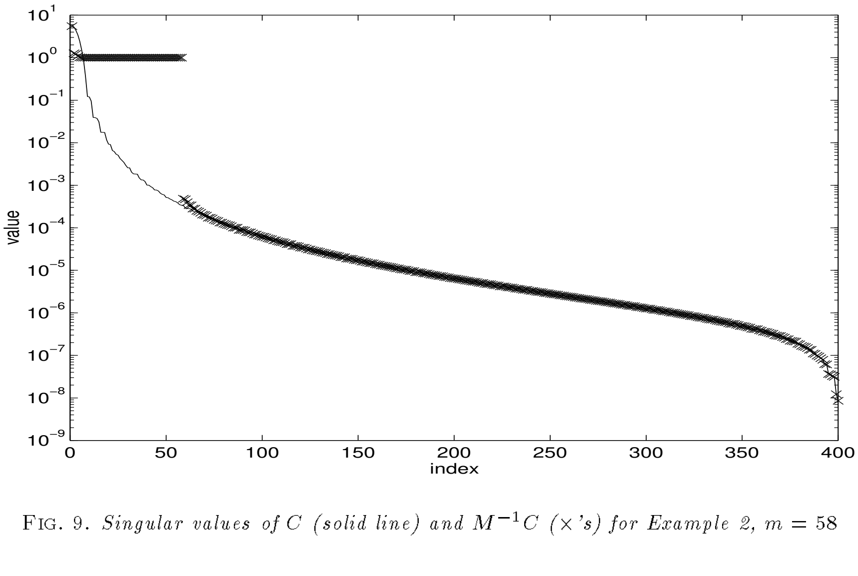

FIG. 9. *Singular values of C (solid line) and $M^{-1}C$ ($\times$'s) for Example 2, $m = 58$*

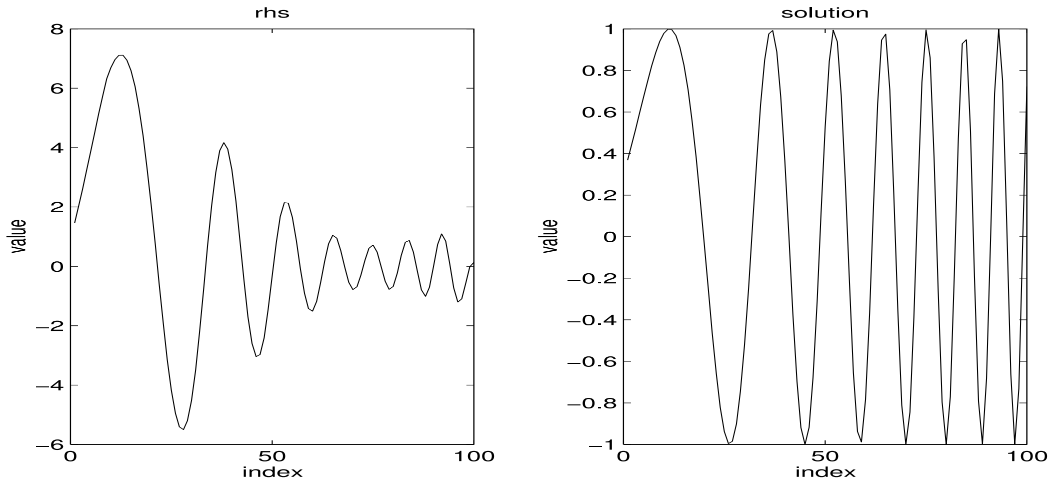

FIG. 10. *Uncontaminated data vector (left) and exact solution vector (right) for Example 3.*

kernel of the integral equation is given by $t(\alpha, \beta) = \phi(\alpha - \beta)$ where ϕ is defined by

$$\phi(\gamma) = \begin{cases} 1 + cos(\frac{\gamma\pi}{3}) & |\gamma| < 3 \\ 0 & |\gamma| \geq 3 \end{cases}$$

and the limits of integration are -6 and 6. We used Hansen's Matlab Regularization Toolbox, described in [15],to generate the corresponding 400×400 symmetric Toeplitz matrix whose condition number was approximately 1×10^8. In this code, the integral equation is discretized by the Galerkin method with orthonormal box functions. The uncorrupted data vector is shown in Figure 6[2]. The noise level was 1×10^{-2} for this problem.

It was difficult to determine the appropriate cutoff value m, as Figure 7 indicates, but Table 2 and Figure 8 show that the savings in the number of iterations to convergence can be substantial. In addition, for several values of m, the minimum relative error is somewhat lower than the minimum obtained for the unpreconditioned problem. For example, after 293 iterations, CGLS on the unpreconditioned problem achieved a minimum relative error of 5.71×10^{-2}. For $m = 58$, however, a minimum relative error of 3.05×10^{-2} was reached in only 9 iterations.

Figure 9 illustrates that, as in Example 1, the first m singular values of the preconditioned matrix are clustered around one and the singular values corresponding to the noise subspace remain almost unchanged.

5.3. Non-symmetric example. Finally, since both previous examples involve symmetric Toeplitz matrices, for our third example we chose to work with the 100×100 matrix T whose first column is defined by

$$\left[\delta^0, \delta^1, \ldots, \delta^9, 0, \ldots, 0\right]^T$$

and whose first row is

$$\left[\delta^0, \delta^1, \delta^{2^2}, \ldots, \delta^{16^2}, 0, \ldots, 0\right]$$

[2] We set $\hat{g} = [s, s, s, s]$ with $s = (\phi(v, -1))^2 + (\phi(v, 1))^2$ where $v = [-5 : 0.1 : 4.9]$ and $\phi(v, \lambda) = \frac{1}{\sqrt{2\pi}} exp(\frac{-(v-\lambda)^2}{2})$. Then $\hat{f}$ was taken to be the exact numerical solution of the problem.

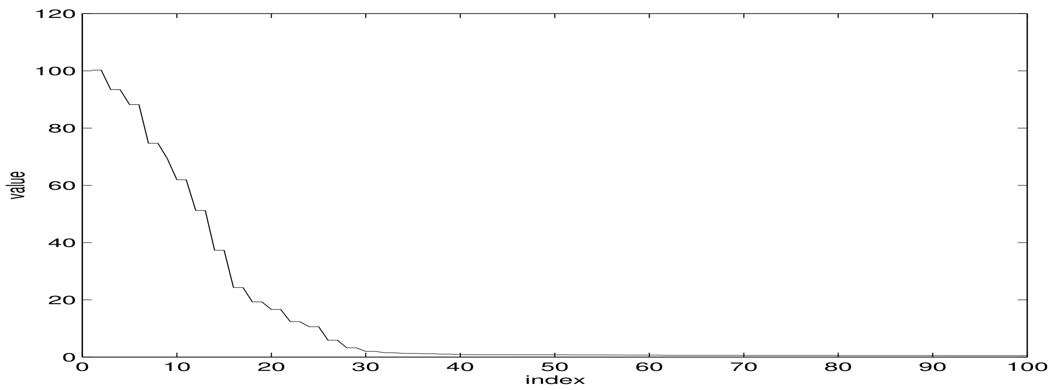

FIG. 11. *Fourier coefficients of the noisy data for Example 3.*

m	minimum rel. error	achieved at iter.
0	2.81×10^{-2}	86
25	2.81×10^{-2}	32
30	2.81×10^{-2}	28
37	2.80×10^{-2}	23
42	2.80×10^{-2}	21
46	2.78×10^{-2}	22
50	2.83×10^{-2}	24
62	3.89×10^{-2}	9

TABLE 3
Minimum relative errors achieved for various values of m, Example 3.

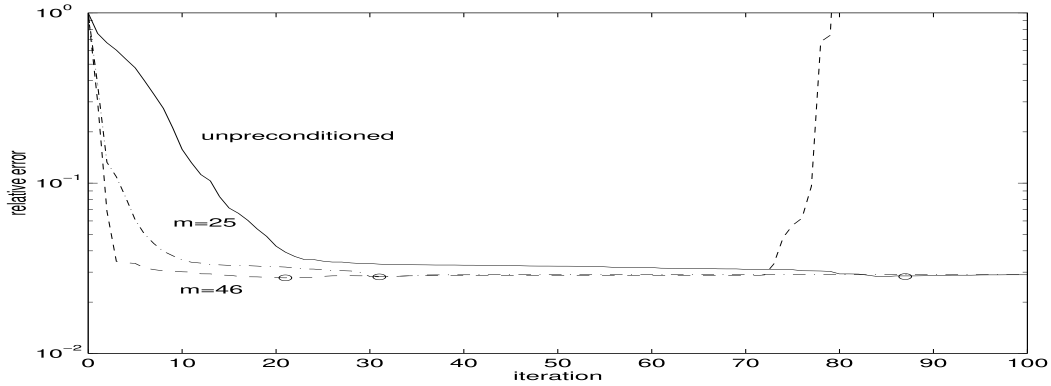

FIG. 12. *Relative error in computed solution for $m = 0$, $m = 25$, and $m = 46$, Example 3.*

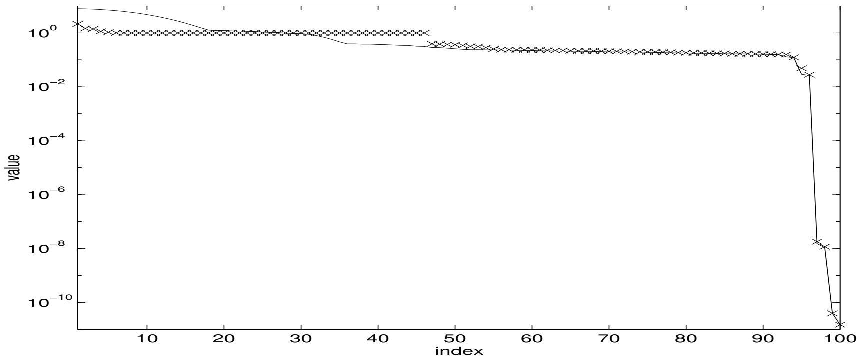

FIG. 13. *Singular values of C (solid line) and $M^{-1}C$ ($\times$'s) for Example 3, $m = 43$.*

with $\delta = .87$

The condition number of T is approximately 5.31×10^{11}, making it the worst conditioned of the three matrices. We first defined the exact solution shown in Figure 10[3]. The uncorrupted data was obtained by calculating $\hat{g} = T\hat{f}$, and is also shown in Figure 10. White noise was added to $\hat{g}$ to obtain the noisy data whose Fourier coefficients are shown in Figure 11, where the noise level was determined to be 1×10^{-3}.

As Figure 12 indicates, the minimum relative error obtained with no preconditioning was 2.13×10^{-1} in 76 iterations. For values of m close to 40, however, the preconditioned system converges in fewer than 10 iterations to the same or better minimum relative error. We also observe from Figure 13 that in addition to clustering the first m singular values around one, preconditioning has the benefit of reducing the condition number.

6. Conclusions. We have developed an efficient algorithm for computing regularized solutions to ill-posed problems with Toeplitz structure. This algorithm makes use of an orthogonal transformation to a Cauchy-like system and iterates using the CGLS algorithm preconditioned by a rank-m partial factorization with pivoting. By exploiting properties of the transformation, we showed that each iteration of CGLS costs only $O(n \lg n)$ operations for a system of n variables.

Our theory predicts that for banded Toeplitz matrices we can expect the preconditioner determined in the course of Gu's fast modified complete pivoting algorithm to cluster the largest singular values of the preconditioned matrix around one, keep the smallest singular values small, and not mix the signal and noise subspaces. Thus CGLS produces a good approximate solution within a small number of iterations. Our results illustrate the effectiveness of our preconditioner for an optimal value of m, and for values in a neighborhood of the optimal one. Hence, our algorithm is both efficient and practical.

Determining the optimal value of m can be difficult, and it appears better to underestimate the value rather than to overestimate it. Advances in computing truly rank-revealing factorizations of Cauchy-like matrices will yield corresponding advances in our algorithm.

[3] $\hat{f} = \sin([1{:}0.1{:}11]^2 \cdot \frac{3\pi}{25})$.

Similar ideas are valid for preconditioners of the form

$$\begin{bmatrix} C_1 & & \\ & C_2 & \\ & & I \end{bmatrix}.$$

where C_1 and C_2 are both Cauchy-like. In practice, C_2 can be determined by computing a partial factorization of the trailing submatrix of C, remaining after C_1 is removed. This method saves time in the precomputation of M but more iterations may be required for convergence.

In future work, we plan to study the use of Cauchy-like preconditioners for two dimensional problems, in which T is block Toeplitz with Toeplitz blocks, and for other matrices related to Cauchy-like matrices.

REFERENCES

[1] A. Bojanczyk, 1996. Personal Communication.

[2] R. Chan, J. Nagy, and R. Plemmons, *Circulant preconditioned Toeplitz least squares iterations*, SIAM J. Matrix Anal. and Appl., 15 (1994), p. 80.

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

[4] T. F. Chan and P. C. Hansen, *A lookahead Levinson algorithm for general Toeplitz systems*, IEEE Proc. Signal Processing, 40 (1992), pp. 1079–1090.

[5] R. D. Fierro, G. H. Golub, P. C. Hansen, and D. P. O'Leary, *Regularization by truncated total least squares*, in Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, J. G. Lewis, ed., Philadelphia, 1994, SIAM Press, pp. 102–105.

[6] T. Finck, G. Heinig, and K. Rost, *An inversion formula and fast algorithms for Cauchy-Vandermonde matrices*, Linear Algebra Appl., 183 (1993), p. 179.

[7] H. E. Fleming, *Equivalence of regularization and truncated iteration in the solution of ill-posed problems*, Linear Algebra Appl., 130 (1990), pp. 133–150.

[8] I. Gohberg, T. Kailath, and V. Olshevsky, *Fast Gaussian elimination with partial pivoting of matrices with displacement structure*, Mathematics of Computation, (1995).

[9] W. Groetsch, *Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*, Pitman Publishing Limited, 1984.

[10] M. Gu, *Stable and efficient algorithms for structured systems of linear equations*, Tech. Report LBL-37690, Lawrence Berkeley Laboratory, 1995.

[11] M. H. Gutknecht and H. M., *Look-ahead Levinson and Schur algorithms for non-Hermitian Toeplitz systems*, Tech. Report IPS 93-11, IPS Supercomputing, ETH-Zurich, 1993.

[12] M. Hanke, *Regularization with differential operators, an iterative approach*, Numerical Functional Analysis and Optimization, 13 (1992).

[13] M. Hanke, J. Nagy, and R. Plemmons, *Preconditioned iterative regularization for ill-posed problems*, Numerical Linear Algebra and Sci. Computing, (1993). L. Reichel, A. Ruttan and R.S. Varga, editors.

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

[15] ———, *Rank Deficient and Discrete Ill-Posed Problems*, PhD thesis, Technical University of Denmark, Lyngby, Denmark, July, 1995. UNIC Report UNIC-95-07.

[16] 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.

[17] G. Heinig, *Inversion of generalized cauchy matrices and other classes of structured matrices*, Linear Algebra in Signal Processing, IMA Volumes in Math. and Appl., 69 (1994), pp. 95–114.

[18] ———, 1996. Personal Communication.

[19] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Standards, 49 (1952), pp. 409–436.

[20] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991.

[21] T. Kailath, S. Kung, and M. Morf, *Displacement ranks of matrices and linear equations*, Journal of Math. Anal. and Appl., 78 (1979), pp. 395–407.

[22] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, Journal of Research of the N.B.S., 49 (1952), pp. 33–53.

[23] D. L. Phillips, *A technique for the numerical solution of certain integral equations of the first kind*, J. Assoc. Comput. Mach., 9 (1962), pp. 84–97.

[24] G. W. Stewart and J. guang Sun, *Matrix Perturbation Theory*, Academic Press, 1990.

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

[26] D. Sweet and R. Brent, *Error analysis of a fast partial pivoting method for structured matrices*, Advanced Signal Processing Algorithms, Proc. of SPIE, 2363 (1995), pp. 266–280.

[27] 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.

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

[29] ———, *Sirt- and CG-type methods for the iterative solution of sparse linear least-squares problems*, Linear Algebra and Appl., 130 (1990), pp. 257–302.

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