Iterative Image Restoration Using Approximate Inverse Preconditioning

James G. Nagy, Robert J. Plemmons, *Member, IEEE,* and Todd C. Torgersen

Abstract—**Removing a linear shift-invariant blur from a signal or image can be accomplished by inverse or Wiener filtering, or by an iterative least-squares deblurring procedure. Because of the ill-posed characteristics of the deconvolution problem, in the presence of noise, filtering methods often yield poor results. On the other hand, iterative methods often suffer from slow convergence at high spatial frequencies. This paper concerns solving deconvolution problems for atmospherically blurred images by the preconditioned conjugate gradient algorithm, where a new approximate inverse preconditioner is used to increase the rate of convergence. Theoretical results are established to show that fast convergence can be expected, and test results are reported for a ground-based astronomical imaging problem.**

I. Introduction

ASTRONOMERS and other scientists have long sought to overcome the degradation of astronomical image quality caused by the effects of atmospheric turbulence. These effects are in part due to the mixing of warm and cold air layers. The resulting "twinkling" of stars and other effects are the main limitations of ground-based imaging for both scientific and defense purposes. As pointed out vividly in recent articles in *Scientific American* [25] and *National Geographic* magazines [31], exciting technological breakthroughs are rapidly coming to the aid of scientists attempting to deblur astronomical images. A fast algorithm for the restoration of such images is the topic considered in this paper.

The improvement in ground-based image quality is now generally attempted in two stages. The first stage occurs as the observed image is initially formed. Specially designed *deformable mirrors* operating in a closed-loop adaptive optics system can partially compensate for the effects of atmospheric turbulence. The systems detect the distortions using either a natural guide star (point) image or a guide star artificially generated from the backscatter of a laser-generated beacon. A wavefront sensor measures the optical distortions, which can then be partially nullified by deforming a flexible mirror in the telescope. To be effective, these corrections have to be performed at real-time speed. Adaptive optics control systems form the subject of considerable recent investigation; see, e.g., [14], [15], and [34].

The second stage of compensating for the effects of atmospheric turbulence occurs generally off-line, and consists of the processing step of *image restoration.* An image partially corrected by the adaptive optics procedure discussed above can generally be enhanced further by off-line computer image restoration. Here, large-scale computations, again using either a natural guide star (point) image or a guide star artificially generated from the backscatter of a laser generated beacon, are used to deconvolve the blurring effects of atmospheric turbulence. In this regard, removing a linear shift-invariant blur from a signal or image can be accomplished by inverse or Wiener filtering, or by an iterative least-squares deblurring procedure [27], [28]. Because of the ill-posed characteristics of the deconvolution problem, in the presence of noise, direct filtering methods often yield poor results [20], [27], [28], [33]; on the other hand, iterative methods often suffer from slow convergence at high spatial frequencies [27], [28]. New fast iterative least squares algorithms have been recently investigated [1], [5]–[8], [23], [30]–[33]. Iterative methods generally can be implemented using less storage than direct methods, and are often less sensitive to ill-conditioning.

This paper concerns solving deconvolution problems for atmospherically blurred images. As in [6]–[8], [22], and [23], we use the preconditioned conjugate gradient iterative algorithm. Here, a new approximate inverse Toeplitz preconditioner is used to further increase the rate of convergence. For the special one-dimensional (1-D) case, the preconditioner is related to an approximation scheme for banded Hermitian positive definite Toeplitz matrices proposed by Jain [26], and further developed as a preconditioner by Linzer [29]. Their scheme, however, requires the solution of a Toeplitz system of equations of dimension the same as the bandwidth, in order to construct their approximation. We avoid this step. In addition, test results for our algorithm are reported for two-dimensional (2-D) imaging problems.

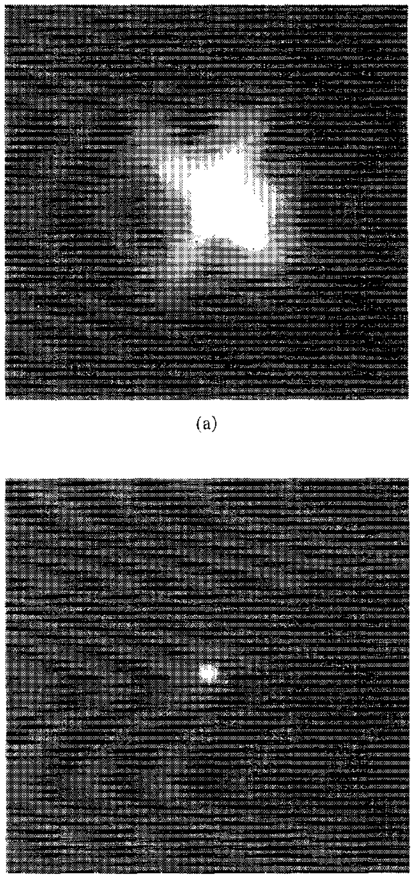
The paper is organized as follows. In Section II we review the main concepts of the preconditioned conjugate gradient (CG) algorithm. In Section III, we consider 1-D deconvolution, where a new Toeplitz approximate inverse preconditioner is introduced and analyzed. Our main interest is in 2-D problems; however, we feel an initial presentation of the 1-D case significantly simplifies the analysis for 2-D problems, which we discuss in Section IV. In Section V we test our algorithms on ground-based astronomical imaging data and provide some concluding remarks.

Manuscript received July 16, 1994; revised October 6, 1995. This work was supported by Oak Ridge Associated Universities under Grant 009707, by the Air Force Office of Scientific Research under Grant F49620-94-1-0261, and by the National Science Foundation under Grant CCR-92-01105. The associate editor coordinating the review of this paper and approving it for publication was Dr. R. L. Lagendijk.

J. G. Nagy is with the Department of Mathematics, Southern Methodist University, Dallas, TX 75275-0156 USA.

R. J. Plemmons and T. C. Torgersen are with the Department of Mathematics and Computer Science, Wake Forest University, Winston-Salem, NC 27109 USA.

Publisher Item Identifier S 1057-7149(96)04543-5.



(a)

(b)

Fig. 1. Observed object and guide star images. (a) Measured image. (b) Guide star image.

II. Preconditioned Conjugate Gradients

In this paper, we solve deconvolution problems using the CG algorithm. To help in describing our techniques, we begin in this section with a brief description of CG and state some important properties relating to it. The CG algorithm is essentially an iterative method that solves Hermitian positive definite systems of equations $Ax = b$, cf. [18]. However, extensions for nonsymmetric, indefinite, and overdetermined least-squares problems exist; see, for instance, Freund *et al.* [17]. For example, in the least squares problem $\min\|g - Hf\|_2$, CG can be applied to the normal equations $H^*Hf = H^*g$ without explicitly forming H^*H. (Here we use the notation * to denote the complex conjugate transpose of a matrix.) Since there are several formulations of CG for least-squares problems, we do not state a particular algorithm here; rather, we refer the reader to [17] for further details.

The convergence of the CG algorithm depends on the eigenvalues of the coefficient matrix A (singular values of H in the least squares problem), cf. Axelsson and Barker [2]. In particular, if the spectrum is clustered around one, then convergence will be rapid. Thus, to make CG a useful iterative method, *preconditioning* is typically used to cluster the spectrum around one. A Hermitian positive definite matrix P is constructed so that $P^{-1}A \approx I$. The CG algorithm is then applied to the system $L^{-1}AL^{-*}y = L^{-1}b$, with $x = L^{-*}y$ and $P = LL^*$. This is done implicitly, with the PCG algorithm stated in Algorithm 1 for a general symmetric positive definite matrix A. (See [18] for further details. An alternate formulation, which sometimes obtains slightly better accuracy, can be found in [3], but the formulation in Algorithm 1 is more appropriate for our deconvolution problem.)

Each iteration of the preconditioned conjugate gradient (PCG) is more expensive than CG, due to the need to solve a linear system with P as the coefficient matrix. However, convergence can be significantly faster for PCG; hence, the overall cost to solve $Ax = b$ can be much less. Moreover, usually one tries to find P so that the application of P is not too expensive. We also remark that often an *approximate inverse preconditioner* can be found. That is, rather than finding P explicitly, a matrix $K = P^{-1}$ is constructed. In this case, rather than solving systems with P, all that is needed is a matrix-vector multiply with K at each iteration. This is the approach we use for deconvolution problems. In our case, however, $A = H^*H$ is not formed explicitly; rather, A is kept in factored form for the least-squares problem. In fact, only a single column of H needs to be used in our implementation. Moreover, K has the form $K = M^*M$, where only a single column of M is needed. (See Section III-B.)

Algorithm 1: PCG [18, Algorithm 10.3.1]

Input
A – The $n \times n$ matrix in $Ax = b$.
b – The $n \times 1$ right hand side.
P – The $n \times n$ preconditioner.

Output
x – The approximate solution to $Ax = b$.

Method
choose an initial approximate solution vector x
$r = b - Ax$
$k = 0$
`while` $\|r\|/\|b\| > tol$
  solve $Pz = r$
  $k = k + 1$
  `if` $k = 1$
    $p = z$
  `else`
    $\beta = r^*z/r^*_{\mathrm{old}}z_{\mathrm{old}}$
    $p = z + \beta p$
  `end`
  $\alpha = r^*z/p^*Ap$
  $x = x + \alpha p$
  $r_{\mathrm{old}} = r$, $z_{\mathrm{old}} = z$
  $r = r - \alpha Ap$
`end.`

III. 1-D Deconvolution

Consider the convolution of two 1-D discrete signals, $g = h \star f$, where f is a vector of length $m = n - \beta$, g is a vector of length n, and the convolution vector h has the form

$$h = [h_0, h_1, \cdots, h_\beta]^T. \tag{1}$$

The convolution can be expressed in matrix notation as $g = Hf$, where H is a $n \times (n-\beta)$ column circulant matrix of the form

$$H = \begin{bmatrix} h_0 & & & \\ h_1 & h_0 & & \\ \vdots & & \ddots & \\ h_\beta & & & \ddots \\ & \ddots & & h_0 \\ & & \ddots & h_1 \\ & & & \vdots \\ & & & h_\beta \end{bmatrix}. \tag{2}$$

In many situations of interest $\beta \ll n$ [27]. In applications such as signal restoration, the observed signal g and the *discrete point spread function* $h = H(:,1)$ (now using h in (1) padded with zeroes to denote the first column of H) are known, and the aim is to compute f. This is known as *deconvolution*.

One common technique used in signal and image processing to compute an approximate solution to the deconvolution problem, known as the inverse filter method, is to implicitly extend H to an $n \times n$ circulant matrix

$$C = [H \quad \hat{H}]$$
$$= \begin{bmatrix} h_0 & & & & & h_\beta & \cdots & h_1 \\ h_1 & h_0 & & & & & \ddots & \vdots \\ \vdots & & \ddots & & & & & h_\beta \\ h_\beta & & & \ddots & & & & \\ & \ddots & & & h_0 & & & \\ & & \ddots & & h_1 & h_0 & & \\ & & & \ddots & \vdots & \vdots & \ddots & \\ & & & & h_\beta & h_{\beta-1} & \cdots & h_0 \end{bmatrix} \tag{3}$$

and solve the system $C\left[\begin{smallmatrix}\tilde{f}\\ \hat{f}\end{smallmatrix}\right] = g$. If C is invertible, H has full column rank, and there is an exact solution to $Hf = g$, then it is easy to see that $\hat{f} = 0$ and $\tilde{f} = f$. The procedure is also called the *extended sequence deconvolution method* [27]. This approach is extremely attractive, since circulant systems of equations can be solved easily in $O(n \log n)$ operations using the fast Fourier transform (FFT), cf. [12]. However, when g is corrupted by noise, it is unlikely that $Hf = g$ has an exact solution. Thus, since H is often ill-conditioned, the computed solution using this technique will be extremely sensitive to the noise [1]. Furthermore, even if $Hf = g$ has an exact solution it may be the case that C is not invertible. For example, if

$$H = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

then

$$C = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

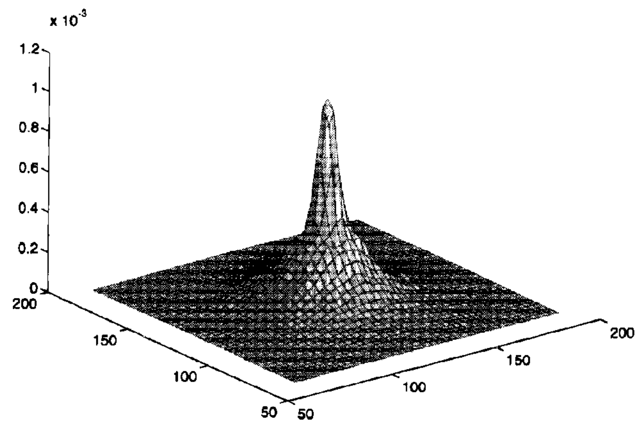


Fig. 2. Surface view of blurring operator.

For this example, we see that C is singular by observing the vector $[1\ 1\ 1\ 1]^T$ is in its nullspace.

We use the PCG algorithm to compute a least-squares solution

$$\min_f \|g - Hf\|_2. \tag{4}$$

This approach is only slightly more expensive than directly inverting C as described above and is still generally an $O(n \log n)$ method. Iterative methods have the advantage of facilitating the imposition of prior information and constraints. They generally can be implemented using less storage than direct methods and are often less sensitive to ill-conditioning. We will show that *regularization*, to compensate for the effects of ill-conditioning in the presence of noise, can easily be incorporated in our approach with essentially no extra work. Therefore, our scheme is less sensitive to noise and can be more accurate than the extended circulant inversion technique described above. We begin by assuming no noise is present and describe an approximate inverse preconditioner for deconvolution. We then describe some implementation details; in particular, we emphasize that the PCG algorithm can be implemented without large storage requirements. The main computational tool we use is the FFT.

A. An Approximate Inverse Preconditioner

In this section, we describe a new approximate inverse preconditioner for deconvolution problems. As noted earlier, for the special 1-D case the preconditioner is related to an approximation scheme for banded Hermitian positive definite Toeplitz matrices proposed by Jain [26] and further developed as a preconditioner by Linzer [29]. Their scheme, however, requires the solution of a Toeplitz system of equations of dimension the same as the bandwidth, in order to construct their approximation, which we avoid as noted below.

Our preconditioner is based on the circulant extension C shown in (3). Since C is circulant, we can factor it as $C = F^* \Lambda F$, where F is the unitary discrete Fourier transform matrix and Λ is a diagonal matrix containing the eigenvalues of C. The eigenvalues of C can be computed as $\sqrt{n} Fc$, where c is the first column of C, using the FFT [12].

If C is nonsingular, then we can easily compute its inverse as $C^{-1} = F^*\Lambda^{-1}F$. Partitioning C^{-1} as

$$C^{-1} = \begin{bmatrix} M^* \\ \hat{M}^* \end{bmatrix} \tag{5}$$

we show later that M^*M can be a good approximate inverse of H^*H. Specifically, we show that M^*M and $(H^*H)^{-1}$ differ only by a matrix of small rank. However, as illustrated earlier, C may not be invertible. In this case, we can still obtain an approximate inverse as follows. Consider the *generalized inverse* of C defined formally as

$$C^\dagger = F^*\Lambda^\dagger F, \quad \text{where}$$
$$[\Lambda^\dagger]_{jj} = \begin{cases} 1/[\Lambda]_{jj} & \text{if } [\Lambda]_{jj} \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

(Notice that if C is nonsingular, then $C^\dagger = C^{-1}$.) We partition $C^\dagger$ as in (5). It will be shown in Theorem 2 that M^*M is still nonsingular, and we again take it as our approximate inverse of H^*H. In any practical implementation, only a single column of M and a single column of H is needed.

Recall that to show M^*M is an effective preconditioner for PCG, we need to show that i) M^*M is positive definite; ii) the eigenvalues of $(M^*M)(H^*H)$ are clustered around one; and iii) matrix-vector multiplication with M^*M can be done efficiently. Since M and M^* are embedded in circulant matrices, iii) is satisfied using the FFT to perform the matrix-vector multiplies, which is elaborated on in Section III-B. For i) and ii) we prove the following theorems.

Theorem 1: Let H be a convolution matrix with full column rank and bandwidth β, and define the circulant matrix C as in (3). If C is nonsingular and M is defined as in (5), then M^*M is positive definite, and

$$(M^*M)(H^*H) = I + R$$

where $\text{rank}(R) \leq \beta$. That is, all but at most β eigenvalues of $(M^*M)(H^*H)$ are precisely equal to one.

Proof: Since C is nonsingular, it follows that the rows of C^{-1} are linearly independent. Thus M has full column rank, and so M^*M is positive definite.

Now, since C is nonsingular

$$(C^*C)^{-1}(C^*C) = \begin{bmatrix} M^* \\ \hat{M}^* \end{bmatrix} [M \quad \hat{M}] \begin{bmatrix} H^* \\ \hat{H}^* \end{bmatrix} [H \quad \hat{H}] = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

which implies

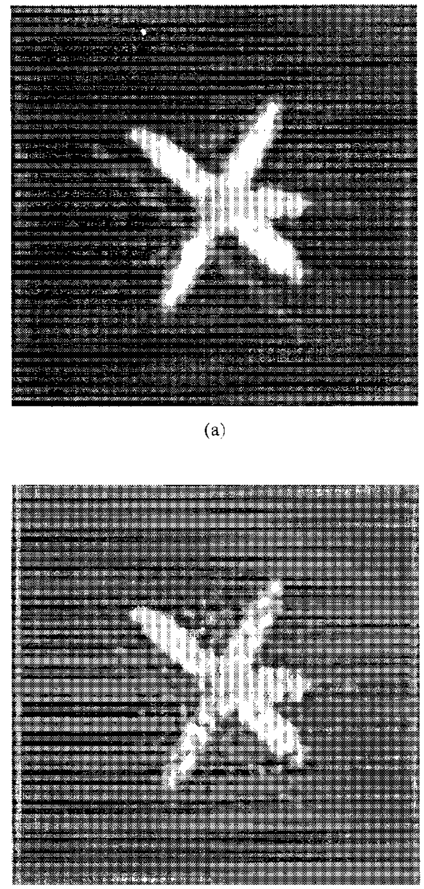
$$\begin{bmatrix} M^*M & M^*\hat{M} \\ \hat{M}^*M & \hat{M}^*\hat{M} \end{bmatrix} \begin{bmatrix} H^*H & H^*\hat{H} \\ \hat{H}^*H & \hat{H}^*\hat{H} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}.$$

Thus

$$(M^*M)(H^*H) + (M^*\hat{M})(\hat{H}^*H) = I.$$

Taking $R = -(M^*\hat{M})(\hat{H}^*H)$, and observing that $\hat{M}$ and $\hat{H}$ are $n \times \beta$ matrices, we see that the rank of R is at most β. □

Theorem 2: Let H be a convolution matrix with full column rank and bandwidth β, and define the circulant matrix C as in (3). If C is singular, with $C^\dagger$ defined as in (6), and M is defined as in (5), then M^*M is positive definite, and



(a)

(b)

Fig. 3. Restored images, no preconditioner, $\mu = 0$. (a) 50 iterations. (b) 100 iterations.

$$(M^*M)(H^*H) = I + R$$

where $\text{rank}(R) \leq 2\beta$. That is, all but at most 2β eigenvalues of $(M^*M)(H^*H)$ are precisely equal to one.

Proof: We first show that M has full column rank, from which it follows that M^*M is positive definite. Suppose x is a nonzero vector of length m such that $Mx = 0$, and observe that $C = CC^\dagger C$ and $(C^\dagger C)^* = C^\dagger C$. Using these relations, we see that

$$\begin{aligned} Hx &= [H \quad \hat{H}] \begin{bmatrix} x \\ 0 \end{bmatrix} = C \begin{bmatrix} x \\ 0 \end{bmatrix} \\ &= CC^\dagger C \begin{bmatrix} x \\ 0 \end{bmatrix} = C(C^\dagger C)^* \begin{bmatrix} x \\ 0 \end{bmatrix} \\ &= CC^*(C^\dagger)^* \begin{bmatrix} x \\ 0 \end{bmatrix} = CC^*[M \quad \hat{M}] \begin{bmatrix} x \\ 0 \end{bmatrix} \\ &= CC^*Mx = 0. \end{aligned}$$

Thus, there is a nonzero x such that $Hx = 0$. This contradicts the fact that H has full column rank and so no such nonzero x exists. We therefore conclude that M must have full column rank; hence M^*M is positive definite.

Now, $(C^*C)^\dagger$ is a generalized inverse of C^*C. Thus,

$$(C^*C)^\dagger(C^*C)(C^*C)^\dagger = (C^*C)^\dagger$$

$\Rightarrow$

$$\begin{bmatrix} M^* \\ \hat{M}^* \end{bmatrix}[M \quad \hat{M}]\begin{bmatrix} H^* \\ \hat{H}^* \end{bmatrix}[H \quad \hat{H}]\begin{bmatrix} M^* \\ \hat{M}^* \end{bmatrix}[M \quad \hat{M}] = \begin{bmatrix} M^* \\ \hat{M}^* \end{bmatrix}[M \quad \hat{M}]$$

$\Rightarrow$

$$\begin{aligned}(M^*M)(H^*H)(M^*M) &+ (M^*\hat{H})(\hat{H}^*H)(M^*M) \\ &+ (M^*M)(H^*\hat{H})(\hat{M}^*M) + (M^*\hat{H}) \\ &\times (\hat{H}^*\hat{H})(\hat{M}^*M) \\ &= M^*M.\end{aligned}$$

Thus, $(M^*M)(H^*H) = I + R$, where

$$\begin{aligned}R = &-(M^*\hat{H})[\hat{H}^*H + (\hat{H}^*\hat{H})(\hat{M}^*M)(M^*M)^{-1}] \\ &- (M^*M)(H^*\hat{H})(\hat{M}^*M)(M^*M)^{-1}\end{aligned}$$

and since $\hat{H}$ has dimensions $n \times \beta$, it follows that $\mathrm{rank}(R) \leq 2\beta$. □

The above theorems show that, regardless of whether C is invertible or not, we obtain convergence of PCG in $O(\beta)$ iterations, and fast convergence when using the approximate inverse preconditioner whenever β is small relative to n. If C is invertible, more eigenvalues of $(M^*M)(H^*H)$ are equal to one than the case when C is singular; hence, faster convergence (in β iterations) will be obtained when C is nonsingular. We apply Tikhonov regularization [19] in our imaging problems, which ensures nonsingularity of C. The conditioning of (M^*M) is addressed in Section III-C. The PCG method can be implemented using only a single column of M and a single column of H, as shown in the next section.

B. PCG for Deconvolution: Implementation Details

In this section, we discuss some implementation details of the PCG algorithm for deconvolution problems. Recall that we are computing a solution to the least-squares problem $\min \|g - Hf\|_2$ and that this can be done by applying the classical PCG algorithm to the factored form of the normal equations $H^*Hf = H^*g$. The computation of PCG is dominated by the matrix vector products M^*Mr and H^*Hp. Since H can be embedded in the circulant matrix C and M can be embedded in the matrix $C^\dagger$, these multiplications can be done using the FFT. For example, consider the multiplication H^*Hp. Since $C = [H \mid \hat{H}]$, we have

$$C^*C = \begin{bmatrix} H^*H & H^*\hat{H} \\ \hat{H}^*H & \hat{H}^*\hat{H}^* \end{bmatrix}.$$

So, $w = H^*Hp$ can be obtained by computing

$$C^*C\begin{bmatrix} p \\ 0 \end{bmatrix} = \begin{bmatrix} H^*Hp \\ * \end{bmatrix} = \begin{bmatrix} w \\ * \end{bmatrix}$$

and stripping off the vector w. Since C is circulant, $C^*C = F^*|\Lambda|^2F$, where Λ is a diagonal matrix containing the eigenvalues of C. As mentioned at the beginning of Section III-A, the eigenvalues of C can be computed by taking the FFT of the first column of C. But because $C = [H \mid \hat{H}]$, the FFT of the first column of C is the FFT of the (zero-padded) discrete point spread function (PSF) h given in (1).

In Algorithm 2, we use the following notation. $\hat{x} = \mathtt{fft}(x)$ is simply the FFT of a vector x. $\circ$ denotes component-wise multiplication. For example, if x and y are two vectors with components x_i and y_i, then $x \circ y$ is the vector with components $x_i y_i$, $i = 1, 2, \cdots, n$. x^c means component-wise complex conjugate of a vector x. That is, if x has components x_i, then x^c is the vector with components $\bar{x}_i$. $x_e = \left[\begin{smallmatrix} x \\ 0 \end{smallmatrix}\right]$, means simply pad zeros at the bottom of a vector x. $1:n$ denotes a sequence of integers from one through n. For example, $x(1:n)$ means the subvector of x containing its first n components. The notation $\hat{y}^{-1}$ denotes the vector whose components are the inverse of the corresponding nonzero components of $\hat{y}$. If $\hat{y}$ has any zero components, then the corresponding components in $\hat{y}^{-1}$ are also taken to be zero.

C. A Regularized PCG Deconvolution Algorithm

Since deconvolution is modeled as an integral equation of the first kind (an ill-posed inverse problem [20], [21]), it is well known that such deconvolution algorithms can be extremely sensitive to noise. This is true for Algorithm 2, in that it is not likely to converge to a solution. Since any realistic problem will have noise, we need to incorporate some form of *regularization* to stabilize the computations. Among the several methods for regularizing ill-posed problems, Tikhonov regularization is probably the best known [19], [21]. This is the regularization we choose to use, since it is easily incorporated into Algorithm 2. In matrix terms, Tikhonov regularization amounts to solving the least squares problem

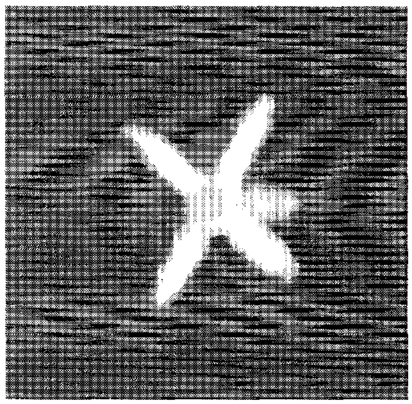
$$\min \left\| \begin{bmatrix} g \\ 0 \end{bmatrix} - \begin{bmatrix} H \\ \mu L \end{bmatrix} f \right\|_2 \tag{7}$$

where L is the regularization operator (sometimes chosen for smoothing purposes) and μ is the regularization parameter that depends on the noise level. The regularization operator, L, is usually chosen to be the identity matrix or some discretization of a differential operator [13]. Thus, the matrix L is often a banded Toeplitz matrix.

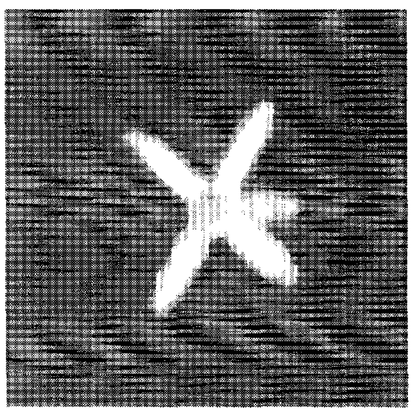
An approximate inverse preconditioner M for this regularized least squares problem can be constructed as follows. Assume that the bandwidth of $L_\mu \equiv \mu L$ is not larger than the bandwidth of H. Embed H into a circulant matrix C_H as was done in Section III-A, and μL into a circulant matrix of the same dimensions by dragging down the diagonals and filling in the northeast and southwest corners to complete the circulant structure. Using ideas similar to those in [7], we now construct a circulant matrix C such that $C^*C = C_H^*C_H + C_L^*C_L$ as follows. Since C_H and C_L are circulant matrices, we can write them as $C_H = F^*\Lambda_H F$ and $C_L = F^*\Lambda_L F$, where Λ_H and Λ_L are diagonal matrices. Then

$$C^*C = F^*(|\Lambda_H|^2 + |\Lambda_L|^2)F.$$

Let $\Lambda = (|\Lambda_H|^2 + |\Lambda_L|^2)^{1/2}$, and define $C = F^*\Lambda F$. Then it is easy to see that C is a real symmetric, nonnegative,



(a)



(b)

Fig. 4. Restored images, no preconditioner. $\mu = 0.018$. (a) 50 iterations. (b) 200 iterations.

definite circulant matrix. Generally, L is nonsingular, so that the resulting C is nonsingular. The approximate inverse preconditioner M is defined as in (5). To indicate that M can be an effective preconditioner, we prove the following theorem.

Theorem 3: Let H be a convolution matrix with full column rank and bandwidth β, L be a regularization operator matrix, and let C and M be defined as above. Then M^*M is positive definite, and $(M^*M)(H^*H+\mu^2L^*L) = I+R$, where

$$\operatorname{rank}(R) \leq \begin{cases} 2\beta & \text{if } C \text{ is nonsingular} \\ 3\beta & \text{if } C \text{ is singular} \end{cases}$$

Proof: It is easy to see that M^*M is positive definite when C is nonsingular. If C

$$C^*C = \begin{bmatrix} H^*H + \mu L^*L + L_{21}^*L_{21} & C_{12} \\ C_{12}^* & C_{22} \end{bmatrix} \tag{8}$$

where $C_{12} = H^*\hat{H} + \mu^2 L^* L_{12} + L_{21}^* L_{22}$ and $C_{22} = \hat{H}^*\hat{H} + L_{12}^* L_{12} L_{22}^* L_{22}$. The upper left submatrix in the above partitioning is positive definite, so it follows from [22, Ths. 2 and 3] that M^*M is positive definite.

To establish the bounds on the rank(R), first consider the partitioning of C^*C given in (8). If C is nonsingular, it follows that

$$\begin{aligned}(C^*C)^{-1}(C^*C) &= \begin{bmatrix} M^*M & M^*\hat{M} \\ \hat{M}^*M & \hat{M}^*\hat{M} \end{bmatrix} \\ &\quad \times \begin{bmatrix} H^*H + \mu^2 L^*L + L_{21}^*L_{21} & C_{12} \\ C_{12}^* & C_{22} \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}.\end{aligned}$$

Thus, we obtain $M^*M(H^*H + \mu L^*L) = I + R$ where

$$R = M^*ML_{21}^*L_{21} + M^*\hat{M}C_{12}^*$$

and rank$(R) \leq 2\beta$.

The proof for the case when C is singular follows easily using arguments similar to those above and to those used in Theorem 2. Therefore, the details are omitted. $\square$

Algorithm 2: PCG for Deconvolution

Input

g – The $n \times 1$ signal vector.
h – The $n \times 1$ convolution vector.

Output

f – The approximate (least squares) solution to
$g = h \star f$.

Method

```
choose an initial approximate solution f
f_e = [f; 0];   ĝ = fft(g);   f̂_e = fft(f_e);
ĥ = fft(h)
ŷ = ĥ^c ∘ ĥ,   r̂_e = ĥ^c ∘ ĝ − ŷ ∘ f̂_e
r_e = ifft(r̂_e);   r = r_e(1:n)
k = 0
while ||r||/||g|| > tol
    r_e = [r; 0];   r̂_e = fft(r_e)
    ẑ_e = ŷ^{-1} ∘ r̂_e
    z_e = ifft(ẑ_e),   z = z_e(1:n)
    k = k + 1
    if k = 1
        p = z
    else
        β = r* z / r*_old z_old
        p = z + βp
    end
    p_e = [p; 0];   p̂ = fft(p_e)
    ŵ_e = ŷ ∘ p̂_e
    w_e = ifft(ŵ_e),   w = w_e(1:n)
    α = r* z / p* w
    f = f + αp
    r_old = r,   z_old = z
    r = r_old − αw
end.
```

We now comment on the implementation details of our regularized PCG algorithm. For simplicity, we take $L = I$ for the remainder of our presentation. The normal equations for the LS problem (7) can then be written as $(H^*H + \mu^2 I)f = H^*g$. Thus, the extended circulant matrices have the form

$$C^*C = C_H^*C_H + \mu^2 I = F^*\left(|\Lambda_H|^2 + \mu^2 I\right)F$$

and so Tikhonov regularization simply amounts to adding μ^2 to each of the eigenvalues of $C_H^*C_H$. In particular, we note that

C^*C is nonsingular. Hence, the smaller bound on the rank(R) in Theorem 3 is obtained. From the above analysis, it follows that the only change to Algorithm 2 needed to incorporate regularization is the computation of $\hat{y}$ in the initialization stage. Specifically, $\hat{y}$ is computed as

$$\hat{y} = \hat{h}^c \circ \hat{h} + \hat{u}, \quad \text{where } \hat{u} = [\mu^2, \mu^2, \cdots, \mu^2]^T.$$

This section is concluded with a discussion of the conditioning of our preconditioner (M^*M). As is well known, the blurring operator matrix H is generally ill-conditioned, since it can have a large number of singular values clustered around zero. Note that we are actually preconditioning $[\,^{H}_{\mu I}]$, which, unless μ is very small, is not ill-conditioned. To obtain a bound for the condition number of M^*M, let λ_1 be the eigenvalue of C_H with largest magnitude, and λ_n be the one with smallest magnitude. Now $C^*C = C_H^*C_H + \mu^2 I$, so the condition number of $(C^*C)^{-1}$ is

$$\kappa((C^*C)^{-1}) = \frac{\lambda_1^2 + \mu^2}{\lambda_n^2 + \mu^2}.$$

Since M^*M is the leading principal submatrix of $(C^*C)^{-1}$, the Cauchy interlace theorem (see [18], Corollary 8.1.4) implies

$$\kappa(M^*M) \leq \kappa((C^*C)^{-1}).$$

Thus, (M^*M) cannot be worse conditioned than the circulant approximation (C^*C) to $H^*H + \mu^2 I$.

IV. 2-D Problems: Atmospheric Blurring

The results of Section III-C extend in a natural way to iterative 2-D image restoration. For notation purposes we assume that the image is n-by-n and thus contains n^2 pixels. Typically, n is chosen to be a power of two, e.g., $n = 256$ or larger. Thus, the number of unknowns is 65 536 or larger.

The image formation process can be modeled as

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y;s,t)f(s,t)\,ds\,dt + \eta(x,y)$$

where $g(x,y)$ is the observed (degraded) image, $f(x,y)$ is the true (original) image (unknown), and $\eta(x,y)$ is assumed to be additive, Gaussian noise. Here, $h(x,y;s,t)$ represents blurring function (PSF) [27], [28].

After discretization and with finite support regions for f and g, a quadrature rule yields the expression

$$g(i,j) = \sum_k \sum_l h(i,j;k,l)f(k,l) + \eta(i,j).$$

By stacking the rows of f and g, one obtains the matrix-equation form

$$g = Hf + \eta$$

where H is the rectangular PSF matrix. This returns us to the least-squares problem (4) for minimizing $\|g - Hf\|_2$ in order to approximate f. In this case, H often has a structured block form that can be used in the process of restoring f.

A. Estimating the PSF h Using a Guide Star

In the spatially invariant case generally assumed for atmospherically blurred images (as well as other applications [27], [28]), the blurring operates uniformly across the image of the object in question, i.e.,

$$h(i,j;k,l) = h(i-k, j-l).$$

Here, H is a banded block column circulant matrix with column circulant blocks. Moreover, the blocks themselves have the banded form shown in (2).

For the 2-D deconvolution problem arising in atmospheric imaging, the imaging system detects the atmospheric distortions using either a natural guide star (point) image or a guide star artificially generated from the backscatter of a laser generated beacon [25]. A wavefront sensor measures the optical distortions, which can then be digitized into a blurred image of the guide star pixel. To form the PSF h, the rows of the blurred pixel image are stacked into a column vector; see, e.g., [28]. The PSF matrix H can then be given in block form

$$H = \begin{bmatrix} H_0 & & & & \\ H_1 & H_0 & & & \\ \vdots & & \ddots & & \\ H_\gamma & & & \ddots & \\ & \ddots & & & H_0 \\ & & \ddots & & H_1 \\ & & & \ddots & \vdots \\ & & & & H_\gamma \end{bmatrix} \tag{9}$$

with h as the first column. Two observations are important here. Since the guide star is blurred across a limited number of pixels, γ is generally much less than the image dimensions. Second, the blocks H_i are column circulant and banded, with the general form (2) with bandwidths β.

Construction of our preconditioner and our deconvolution Algorithm 3 RPCG for the 2-D case proceeds as in the 1-D case, with the exception that now the 2-D FFT is used instead of the 1-D FFT. Here, the matrix H is extended to a block circulant matrix with circulant blocks, where each block has the form (3). The preconditioner M^*M and regularization procedure are derived in a manner similar to that used in Section III.

We begin by giving a 2-D version of the approximate inverse preconditioner result given in Theorem 1. In the 2-D case, the preconditioner C in conjunction with an image with n^2 pixels is an $n^2 \times n^2$ matrix obtained by padding columns of each block H_i and then padding column blocks of H. Therefore, the column partitioning $C = [H \mid \hat{H}]$ given by (3) in the 1-D case is not adequate for expressing the desired extension of H in the 2-D case. However, the extension of H may be properly described by

$$CP = [H \mid \hat{H}] \quad \Leftrightarrow \quad C = [H \mid \hat{H}]P^* \tag{10}$$

and

$$P^*C^{-1} = \begin{bmatrix} M^* \\ \hat{M}^* \end{bmatrix} \quad \Rightarrow \quad C^{-1} = P\begin{bmatrix} M^* \\ \hat{M}^* \end{bmatrix} \tag{11}$$

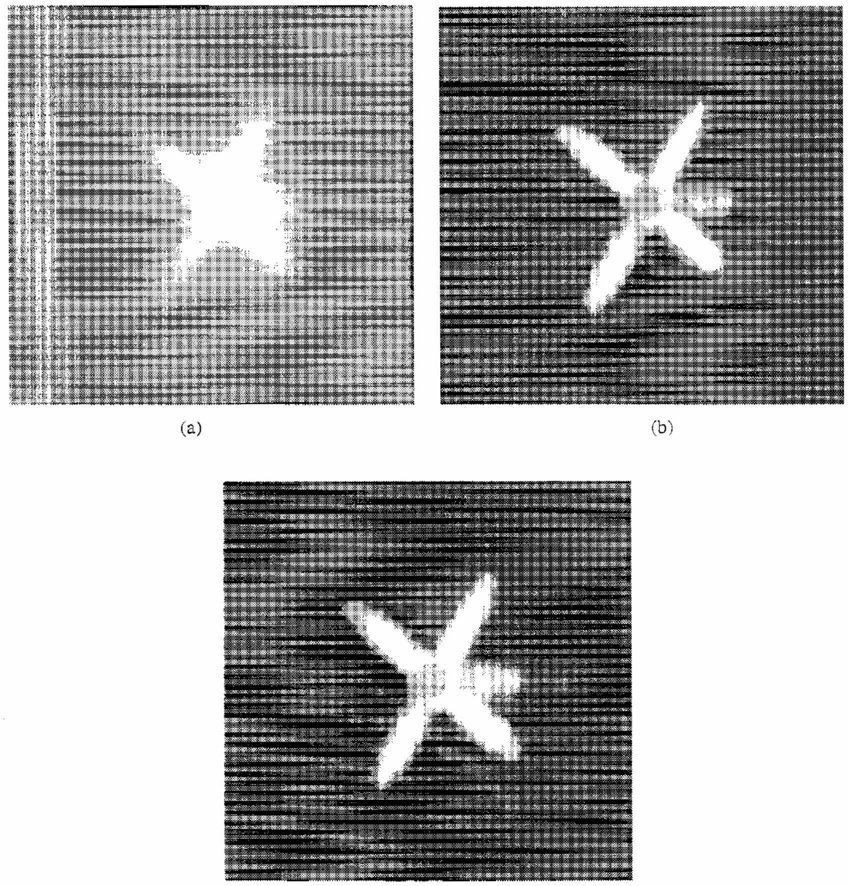


Fig. 5. Restored image, our preconditioner, $\mu = 0.018$. (a) One iteration. (b) Two iterations. (c) Three iterations.

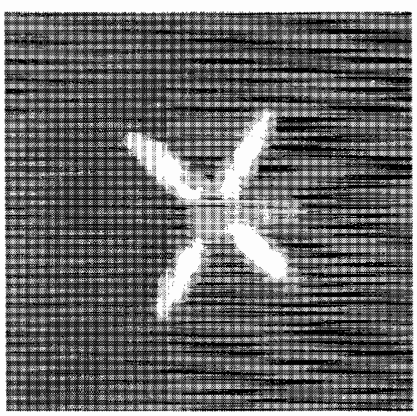


Fig. 6. Restored image, our preconditioner, $\mu = 0.018$ (four iterations).

where P is a permutation matrix acting on the columns of C. The following theorem extends Theorem 1 to the 2-D case.

Theorem 4: Consider an image with n^2 pixels. Let H be a 2-D convolution matrix given in (9) with full column rank where H has column block bandwidth γ and each block H_i has bandwidth β. Let C and M be defined as in (10) and (11), respectively. If C is nonsingular, then M^*M is positive definite, and

$$(M^*M)(H^*H) = I + R$$

where $\text{rank}(R) \leq n\gamma + n\beta - \beta\gamma$.

Proof: By arguments similar to those presented in Theorem 1, M^*M is positive definite. To establish the desired bound on $\text{rank}(R)$, observe that since C is nonsingular

$$C^*C = P\begin{bmatrix} H^* \\ \hat{H}^* \end{bmatrix}[H \mid \hat{H}]P^* = P\begin{bmatrix} H^*H & H^*\hat{H} \\ \hat{H}^*H & \hat{H}^*\hat{H} \end{bmatrix}P^*$$

and

$$(C^*C)^{-1} = P\begin{bmatrix} M^* \\ \hat{M}^* \end{bmatrix}[M \mid \hat{M}]P^* = P\begin{bmatrix} M^*M & M^*\hat{M} \\ \hat{M}^*M & \hat{M}^*\hat{M} \end{bmatrix}P^*.$$

Thus

$$(C^*C)^{-1}(C^*C) = P\begin{bmatrix} (M^*M)(H^*H) + (M^*\hat{M})(\hat{H}^*H) & * \\ * & * \end{bmatrix}P^* = I$$

and so

$$\begin{bmatrix} (M^*M)(H^*H) + (M^*\hat{M})(\hat{H}^*H) & * \\ * & * \end{bmatrix} = I. \tag{12}$$

Equation (12) is the same relationship obtained in the proof of Theorem 1. Thus, taking $R = -(M^*\hat{M})(\hat{H}^*H)$ and observing that $\hat{M}$ and $\hat{H}$ are $n^2 \times (n\gamma + n\beta - \beta\gamma)$ matrices, we have $\text{rank}(R) \le n\gamma + n\beta - \beta\gamma$. □

From the above analysis, it is not difficult to see that analogs of Theorems 2 and 3 can be obtained, keeping in mind the dimensions of $\hat{M}$ and $\hat{H}$ for the block case. In particular, the analog of Theorem 2 for the block case has bounds

$$\text{rank}(R) \le 2(n\gamma + n\beta - \beta\gamma).$$

We next state our 2-D version of the 1-D singular value clustering and regularized PCG convergence result given in Theorem 3. The proof follows along the lines of the proof of Theorem 3 and is omitted for brevity.

Theorem 5: Consider an image with n^2 pixels. Let H be a convolution matrix given in (9) with full column rank, where H has column block bandwidth γ and each block H_i has column bandwidth β. Let L be a block Toeplitz with Toeplitz blocks regularization operator matrix (e.g., as in [28]), and let C and M be defined as discussed above. Then M^*M is positive definite, and $(M^*M)(H^*H + \mu^2 L^*L) = I + R$, where

$$\text{rank}(R) \le \begin{cases} 2(n\gamma + n\beta - \beta\gamma) & \text{if } C \text{ is nonsingular} \\ 3(n\gamma + n\beta - \beta\gamma) & \text{if } C \text{ is singular.} \end{cases}$$

It should be pointed out that Theorems 4 and 5 give very pessimistic bounds on the number of iterations required for convergence of our scheme. The theorems do not take into account possible clustering of the singular values not equal to one, around one. Such tight clustering is illustrated in the computations reported in Section V for atmospheric deblurring. Here, the guide star blurring operator pixel values decay away from the point source, as indicated in Fig. 2.

We further mention that other preconditioning schemes are possible for image restoration; see, for example, Chan and Ng [9]. Perhaps the most popular of these is the optimal circulant preconditioner proposed by Chan and Olkin [10]. In [8], it is shown that this preconditioner can be effective for image restoration problems. Indeed, because it has several desirable properties, the optimal preconditioner should be the preferred choice for general matrices H with large bandwidths. However, as is the case in many image restoration problems, if the region of support of the PSF is much smaller than the dimensions of the image (i.e., small bandwidth), then the approximate inverse preconditioner proposed in this paper can be significantly better. Specifically, [8, Th. 1] (which is analogous to our Theorem 4) shows that the optimal preconditioner can have up to twice as many eigenvalues not clustered around one; that is, PCG can take up to twice as many iterations using the optimal circulant preconditioner [10] than it does using our approximate inverse approach.

We now consider the 2-D regularized preconditioned conjugate gradient algorithm (Algorithm 3: 2-D RPCG) for finding f such that

$$f = \text{argmin}\|g - Hf\|_2. \tag{13}$$

Since H is an $n^2 \times (n-\beta)(n-\gamma)$ matrix, (13) naturally defines f to represent a smaller image than g. In order to facilitate visual comparison of f and g, the support of g and H can be extended to $(n+\beta) \times (n+\gamma)$ so that an $n \times n$ solution is computed. For objects against a dark background, it is reasonable to assume that g and the PSF are zero in the area of extended support.

Throughout the algorithm, multiplications by the inverse preconditioner M^*M and the normal equations matrix H^*H in factored form are accomplished using the 2-D discrete Fourier transform and convolution theorem. In the context of a 2-D convolution, it is advantageous to treat the first column of H as an $n \times n$ matrix h; the known image g and the unknown image f are also treated as an $n \times n$ matrix and $m \times m$ matrix, respectively. Each matrix is padded with sufficient zeros to ensure that component-wise multiplication in the Fourier domain corresponds to a nonwrapped convolution. An adjustment is made to the order of the rows and columns of h to ensure that the desired portions of the convolutions appear in the $n \times n$ leading principle submatrix after returning to the spatial domain. The known (blurred) image is used as the initial approximate solution.[1] For simplicity, we present the algorithm in the case where $\beta = \gamma$.

The notation $a \circ b$ denotes element-wise multiplication and $a \not\circ b$ denotes element-wise division. The function sum2() sums all the elements of a matrix and the function fft2() is the standard 2-D discrete Fourier transform. The function rotate() reorders the rows and columns of a matrix as follows: let $b = \text{rotate}(a, k)$, where a is an $n \times n$ matrix, then

$$b_{i,j} = a_{((i+k-1)\,\text{mod}\,n)+1,((j-k-1)\,\text{mod}\,n)+1}.$$

Scalars in mixed dimensional expressions (e.g., a matrix plus a scalar) are added element-wise in the usual way.

Algorithm 3: 2-D RPCG for Deconvolution

Input

g – An $n \times n$ matrix representing a known (blurred) image.

h – An $n \times n$ matrix representing a blurring operator. h is the (blurred) image of a point centered in the field of view.

β – A nonnegative integer giving the bandwidth of the blurring operator.

μ – A nonnegative Tikhonov regularization parameter.

Output

f – An ($m \times m$ matrix) approximate solution to $g = h \star f$.

Method

$f_e = \begin{bmatrix} f & 0 \\ 0 & 0 \end{bmatrix}$, f_e is an $n \times n$ matrix

$h_r = \text{rotate}(h, n/2)$

$\hat{g} = \texttt{fft2}(g)$, $\quad \hat{h} = \texttt{fft2}(h_r)$

$\hat{f}_e = \texttt{fft2}(f_e)$, $\quad \hat{y} = \hat{h}^c \circ \hat{h} + \mu^2$

$\hat{r}_e = \hat{h}^c \circ \hat{g} - \hat{y} \circ \hat{f}_e$

$r_e = \texttt{ifft2}(\hat{r}_e)$, $\quad r = r_e(1:m, 1:m)$

$k = 0$

[1] Starting CG with a nonzero initial approximation x requires a little more work to compute the initial residual $r = H^*b - H^*Hx$ than starting with $x = 0$. In our application, using the blurred image as the initial approximate requires only one extra component-wise multiplication and one component-wise subtraction; no additional FFT's are necessary.

while $\|r\|_F/\|b\|_F > tol$
  $r_e = \begin{bmatrix} r & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{r}_e = \mathtt{fft2}(r_e)$
  $\hat{z}_e = \hat{r}_e \oslash \hat{y}$
  $z_e = \mathtt{ifft2}(\hat{z}_e), \quad z = z_e(1:m, 1:m)$
  $\rho = \mathrm{sum2}\ (r \circ z)$
  $k = k + 1$
  if $k = 1$
   $p = z$
  else
   $\beta = \rho/\rho_{\mathrm{old}}$
   $p = z + \beta p$
  end
  $p_e = \begin{bmatrix} p & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{p} = \mathtt{fft2}(p_e)$
  $\hat{w}_e = \hat{y} \circ \hat{p}_e$
  $w_e = \mathtt{ifft2}(\hat{w}_e), \quad w = w_e(1:m, 1:m)$
  $\alpha = \rho/\mathrm{sum2}(p \circ w)$
  $f = f + \alpha p$
  $\rho_{\mathrm{old}} = \rho$
  $r = r - \alpha w$
end.

V. Computational Results

In this section, we use Algorithm 3 to remove blur due to atmospheric turbulence. The data was provided to us by R. Carreras [4] at the Lasers and Imaging Directorate, Phillips Air Force Laboratory, Kirtland Air Force Base, NM, USA. The problem consists of a 256×256 image of an ocean reconnaissance satellite observed by a simulated ground-based imaging system, together with a 256×256 image of a guide star observed under similar circumstances. Carreras used computer simulations to obtain the data. Both images are illustrated in Fig. 1. Fig. 2 illustrates the guide star as a surface plot.

Figs. 1 and 2 suggest that most significant information in the estimate of the blurring operator is confined to a central region approximately 1/8 to 1/4 of the total image area. The localized effect of the blurring operator may be exploited to reduce the length of the convolution vector h; a number of approaches are possible. Theorem 5 motivates an approach in which the estimate of the blurring operator is truncated by setting it to zero outside a $\beta \times \gamma$ central region for suitably chosen β and γ. Unfortunately, truncating the blurring operator produces undesirable artifacts (light/dark bands) in the restored image for this example.

Alternatively, all available data for the blurring operator can be used in an "undersized" convolution vector, which allows the small values near the edges to wrap around during the FFT-based computation of the necessary convolutions. Essentially, a (partially) wrapped convolution can be used to approximate a nonwrapped convolution.

Both approaches produce acceptable restorations for this example, but to present the best quality restoration in the remaining discussion, we choose the former approach and use $\beta = \gamma = 255$. In light of Theorem 5, it is surprising that the number of iterations is still quite small and convergence with a good restoration is achieved. We attribute this result to the fact that the data values outside a small region in the blurring operator are sufficiently close to—but not precisely—zero (recall Fig. 2). In particular, the guide star blurring operator pixel values decay away from the point source, as indicated in Fig. 2. From Figs. 5 and 6 we see how this apparent effect of clustering the spectrum around one results in rapid convergence.

We compare the success of the CG algorithm under four circumstances: with and without preconditioning, and with and without regularization, as shown in Figs. 3–6. In the regularized case, trial restorations and inspection are used to determine a visually "best" value for the regularization parameter $\mu = 0.018$.

With no preconditioner and no regularization, the successive approximate solutions improve rather slowly for a number of iterations and subsequently degenerate. The best restoration is achieved in approximately 50 iterations and is illustrated along with a degenerate image in Fig. 3.

The degeneration of the image over iterates beyond 50 does not correspond to a rise in the norm of the normal equations residual $\|r\|_2 = \|H^*b - H^*Hx\|_2$. The behaviors of the approximate solutions and residuals appear to be entirely due to ill conditioning in the deconvolution problem. Since the blurring operator H is nearly rank deficient, the set $\{y \mid H^*Hy = 0\}$ is "nearly a nonzero subspace." Thus, the set of approximate solutions $\{x + y \mid H^*Hx = H^*b, H^*Hy \approx 0\}$ contains undesirable restorations as well as desirable ones.

With no preconditioning but regularizing with Tikhonov parameter $\mu = 0.018$, the successive approximate solutions behave almost identically to the previous case. Improved conditioning due to regularization is not sufficient to accelerate convergence. However, successive approximate solutions of the regularized system do not degenerate as the number of iterations increases. These observations are illustrated in Fig. 4.

Using the preconditioner described in Section III-A but no regularization ($\mu = 0$), we found the successive approximate solutions degenerate into a dull gray in just one iteration and do not improve in subsequent iterations. This result may be attributed to ill conditioning in both the preconditioner and H and agrees precisely with the analysis given in [23].

Finally, in Figs. 5 and 6 we present restorations using the preconditioner described in Section III-A and Tikhonov regularization parameter $\mu = 0.018$. An acceptable restoration is achieved in just three iterations, and very little change occurs beyond five iterations.

The computational effort required with the preconditioner (three iterations) is roughly 3.7×10^8 floating point operations instead of 2.6×10^9 floating point operations without the preconditioner (50 iterations).[2] The resulting speedup factor is approximately seven.

We note that for this particular example, the region of support of the point spread function (PSF) (see Fig. 2) is fairly large. In light of the remarks near the end of Section IV, we might expect the optimal preconditioner proposed by Chan and Olkin [10] to perform better than our approximate

[2] Preconditioning requires one extra forward and one extra backward FFT per iteration.

inverse preconditioner. However, this was not the case. We applied PCG with the optimal preconditioner, using the regularization parameter $\mu = 0.018$. Because the exact solution is available, we were able to compute the true relative errors at each iteration. The best solution we obtained using the optimal circulant preconditioner from [10] and [8] was at five iterations, with a relative error of 0.4110. Our approximate inverse preconditioner, on the other hand, obtained its best solution at four iterations with a relative error of 0.4043.

A. Comments and Future Work

- In the computations described above, our preconditioner is effective for accelerating convergence of the conjugate gradient algorithm applied to image restoration problems, provided regularization is used.
- For atmospheric deblurring, the guide star blurring operator pixel values decay away from the point source, as indicated in Fig. 2. From Fig. 5, we see how this apparent effect of clustering the spectrum around one results in rapid convergence, i.e., convergence in only five iterations for a problem with over 65 000 unknowns. A formal convergence analysis based on this clustering effect for iterative deconvolution problems in atmospheric blurring would be useful and is a topic for future work.
- In this paper, we have used Tikhonov regularization. Other formal regularization techniques should be compared with this approach. For example, L-curve methods [21], [24], regularization based on truncating the singular values of the preconditioner [23], and total variation methods [30], [33] may be appropriate for the problems considered here.
- Recently a Lanczos-based iterative method with regularization has been proposed for ill-posed problems [16]. The method, which is known as the *errors in variables least-squares model* in the statistics literature and *total least squares* in the mathematics literature, would enable consideration of the errors in the coefficient matrix H as well as the observed image g in (13). This is precisely the situation we have in the astronomical imaging problem, where the blurring operator H is estimated using either a natural guide star (point) image or a guide star artificially generated from the backscatter of a laser-generated beacon. We plan to study the application of methods in [16] in our preconditioned iterative deconvolution computations.
- The first two authors and M. Hanke proposed a regularized PCG method in [23]. An effort is now being made to apply the scheme in [23] to the development of an adaptive image restoration scheme using multilevel preconditioning. This approach will be compared with the approximate inverse preconditioner given herein.

ACKNOWLEDGMENT

The authors thank R. Carreras and B. Ellerbroek from the Lasers and Imaging Directorate, Phillips Laboratory, Kirtland Air Force Base, for providing data and encouragement for this project. They also thank A. Nachman from the AFOSR for initiating contact with the Phillips Laboratory researchers. Finally, they are grateful to the referees for their careful reading of the original manuscript, as well as their helpful suggestions for improving the paper.

REFERENCES

[1] J. Abbiss, J. Allen, R. Bocker, and H. Whitehouse, "Fast regularized deconvolution in optics and radar," in *Proc. 3rd IMA Conf. Math. Signal Processing*, 1994.

[2] O. Axelsson and V. Barker, *Finite Element Solution of Boundary Value Problems, Theory and Computation.* New York: Academic, 1984.

[3] A. Bjorck, "least squares methods," in *Handbook of Numerical Methods*, vol. 1, P. Ciarlet and J. Lions, Eds. Amsterdam: Elsevier/North Holland, 1989.

[4] R. Carreras, personal correspondence, 1993.

[5] R. Carreras and M. Roggemann, "Image restoration using nonlinear optimization techniques for partially compensated imaging systems," in *Proc. SPIE Symp. Inverse Problems Scattering Imaging*, San Diego CA, 1992, vol. 1767.

[6] J. Chalidabhongse and J. C. C. Kuo, "A preconditioning technique for fast iterative image restoration," in *Proc. IEEE ICCAS*, San Diego, CA, 1992, pp. 10–13.

[7] R. Chan, J. Nagy, and R. Plemmons, "Circulant preconditioned Toeplitz least squares iterations," *SIAM J. Matrix Anal.*, vol. 15, pp. 80–97, 1994.

[8] ——, "FFT-based preconditioners for Toeplitz-block least squares problems," *SIAM J. Numer. Anal.*, vol. 30, pp. 1740–1768, 1993.

[9] R. Chan and M. Ng, "Conjugate gradient methods for toeplitz systems," preprint, 1995.

[10] T. Chan and J. Olkin, "Preconditioners for Toeplitz-block matrices," *Numer. Alg.*, vol. 6, pp. 89–101, 1993.

[11] N. Clinthorne *et al.*, "Preconditioning methods for improved convergence rates in iterative reconstructions," *IEEE Trans. Med. Imaging*, vol. 12, pp. 78–83, 1993.

[12] P. Davis, *Circulant Matrices.* New York: Wiley, 1979.

[13] L. Eldén, "An algorithm for the regularization of ill-conditioned, banded least squares problems," *SIAM J. Sci. Stat. Comput.*, vol. 5, pp. 237–254, 1984.

[14] B. L. Ellerbroek, "First-order performance evaluation of adaptive optics systems for atmospheric turbulence compensation in extended field-of-view astronomical telescopes," *J. Optic. Soc. Amer.*, vol. 11A, pp. 783–805, 1994.

[15] B. L. Ellerbroek, C. Van Loan, N. P. Pitsianis, and R. J. Plemmons, "Optimizing closed-loop adaptive optics performance using multiple control bandwidths," *J. Optic. Soc. Amer.*, vol. 11A, pp. 2871–2886, 1994.

[16] R. D. Fierro, G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Regularization by truncated total least squares," preprint, 1994.

[17] R. Freund, G. Golub, and N. Nachtigal, "Iterative solution of linear systems," *Acta Numer.*, vol. 1, pp. 57–100, 1992.

[18] G. Golub and C. van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: The Johns Hopkins University Press, 1989.

[19] C. Groetsch, *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind.* Boston, MA: Pittman, 1984.

[20] ——, *Inverse Problems in the Mathematical Sciences.* Wiesbaden, Germany: Vieweg, 1993.

[21] M. Hanke and P. C. Hansen, "Regularization methods for large-scale problems," *Surveys Math. Indust.*, vol. 3, pp. 253–315, 1993.

[22] M. Hanke and J. Nagy, "Toeplitz approximate inverse preconditioner for banded Toeplitz matrices," *Numer. Algeb.*, vol. 7, pp. 183–199, 1994.

[23] M. Hanke, J. Nagy, and R. Plemmons, "Preconditioned iterative regularization for ill-posed problems," in *Numerical Linear Algebra and Scientific Computing*, L. Reichel, A. Ruttan, and R. S. Varga, Eds. Berlin: de Gruyterpp, 1993, pp. 141–163.

[24] 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.*, pp. 1487–1503, 1993.

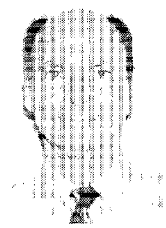
[25] J. W. Hardy, "Adaptive optics," *Sci. Amer.*, vol. 270, no. 6, pp. 60–65, 1994.

[26] A. K. Jain, "Fast inversion of banded Toeplitz matrices by circular decomposition," *IEEE Trans. on Acoust., Speech and Signal Proc.*, vol. 26, pp. 121–126, 1978.

[27] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[28] R. L. Lagendijk and J. Biedmond, *Iterative Identification and Restoration of Images.* Boston, MA: Kluwer, 1991.

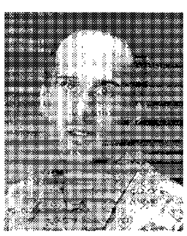
[29] E. Linzer, "Extended circulant preconditioning of toeplitz systems," preprint, IBM Watson Res. Center, Oct. 1992.
[30] L. Rudin, S. Osher, and E. Fatemi, "Total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.
[31] R. A. Smith, "Seeing the universe through new eyes," *Nat. Geograph.*, vol. 185, no. 1, pp. 2–41, 1994.
[32] B. J. Sullivan and A. K. Katsaggelos, "New termination rule for linear iterative image restoration," *Optic. Eng.*, vol. 29, pp. 471–475, 1990.
[33] C. Vogel and M. Oman, "Iterative methods for total variation denoising," *SIAM J. Sci Comput.*, vol. 17, pp. 227–238, 1996.
[34] B. M. Welsh and C. S. Gardner, "Effects of turbulence-induced anisoplanatism on the imaging performance of adaptive-astronomical telescopes using laser guide stars," *J. Optic. Soc. Amer.*, vol. 8A, pp. 69–80, 1991.



Robert J. Plemmons (M'92) received the B.S. degree in mathematics from Wake Forest University, Winston-Salem, NC, in 1962, and the Ph.D. degree in applied mathematics from Auburn University, Auburn, AL, in 1965.

He was on the faculty of the University of Tennessee, Knoxville, from 1966 to 1980, serving a term as Head of Computer Science, and at North Carolina State University, Raleigh, NC, from 1981 to 1991. There, he helped to found the UNC System Center for Research in Scientific Computation in 1988, and served as Director of the Center until 1991 before accepting a research professorship at Wake Forest University. He is currently Z. Smith Reynolds Professor of Mathematics and Computer Science at Wake Forest University in Winston-Salem, NC, and is also Adjunct Professor in computer science at North Carolina State University. His nonacademic positions include minor league baseball with the Baltimore Orioles for four years, research positions at Martin-Marietta Co. and the National Security Agency, and consulting positions with the Oak Ridge National Laboratory and with General Electric CRD. He currently works in scientific computation and its applications to signal and image processing, and in parallel algorithm design.

Dr. Plemmons is the author of more than 120 research papers and three books; he serves on the editorial boards of four journals. He has directed 26 M.S. students and 15 Ph.D. students. He is a member of OSA, SIAM, and SPIE.



James G. Nagy received the B.S. and M.S. degrees in mathematics from Northern Illinois University, DeKalb, IL, in 1986 and 1988, respectively, and the Ph.D. degree in mathematics from North Carolina State University, Raleigh, NC, in 1991.

In 1991–1992, he was a postdoctoral research fellow with the Institute for Mathematics and Its Applications at the University of Minnesota, Minneapolis. In 1992, he joined the Mathematics Department at Southern Methodist University, Dallas, TX, where he is currently an assistant professor. His research interests include numerical solution to structured discrete ill-posed problems, and scientific computation applications in signal and image processing.

Todd C. Torgersen received the B.S. degree in mathematics and the M.S. degree in mathematics from Syracuse University, NY, in May 1975 and May 1977, respectively. After serving as an instructor at Rowan College from 1980 to 1985, he left to study computer science at the University of Delaware. He completed the Ph.D. degree in computer science in May 1989.

From August 1989 to May 1995 he served as Dana Assistant Professsor of Computer Science at Wake Forest University in Winston-Salem, NC. In June 1995, he was promoted to his current position of associate professor of computer science at Wake Forest University. His current research interests include parallel algorithms, symbolic computation, and signal and image processing.

Dr. Torgersen is a member of ACM and SIAM.