SIAM J. MATRIX ANAL. APPL.
Vol. 24, No. 4, pp. 1018–1037

BLIND DECONVOLUTION USING A REGULARIZED STRUCTURED TOTAL LEAST NORM ALGORITHM*

ARMIN PRUESSNER† AND DIANNE P. O'LEARY‡

Abstract. Rosen, Park, and Glick proposed the structured total least norm (STLN) algorithm for solving problems in which both the matrix and the right-hand side contain errors. We extend this algorithm for ill-posed problems by adding regularization, and we use the resulting algorithm to solve blind deconvolution problems as encountered in image deblurring when both the image and the blurring function have uncertainty. The resulting regularized structured total least norm (RSTLN) algorithm preserves any affine structure of the matrix and minimizes a discrete ℓ_p-norm measure of the error, where $p = 1$, 2, or ∞. We demonstrate the effectiveness of these algorithms for blind deconvolution.

Key words. least squares, total least squares, total least norm, structured total least norm, minimization, regularization, ill-posed problem, 1-norm, 2-norm, ∞-norm, overdetermined linear system, blind deconvolution, image deblurring, boundary conditions, constrained total least squares

AMS subject classifications. 65F22, 65K10, 90C05

PII. S0895479801395446

1. Introduction and background.

Most image recording devices fail to record the intensity of a given image scene exactly. Each recorded image section (or pixel) describing the corresponding scene has errors in the form of either random noise, or blurring, or both. *Blurring* occurs when the recorded intensity of a given pixel is in effect influenced by the intensity of neighboring sections. Because of these imperfections in recorded images, it is often necessary to apply deblurring techniques to obtain clearer images.

The problem of image deblurring [6, 11] is modeled as an integral equation of the first kind,

$$\int_\Omega a(s,t)x(t)\ dt = b(s) - r(s) = b_c(s), \tag{1.1}$$

where $s, t \in \mathbf{R}^2$ are the spatial coordinates, Ω the domain or (nonzero) support of the image, $x : \mathbf{R}^2 \to \mathbf{R}$ the true image, $a : \mathbf{R}^4 \to \mathbf{R}$ the point spread function, and $r : \mathbf{R}^2 \to \mathbf{R}$ random noise. The function $b(s)$ is the observed, blurred, noisy image, and $b_c(s)$ is the noiseless blurred image.

In particular, if it is assumed that $a(s,t)$ is *spatially invariant*, that is, its effect depends only on the spatial distance between s and t, then the preceding equation represents a convolution integral, where $a(s,t) = a(s-t)$. In this case, $b_c(s)$ is the result of convolving $a(s)$ and $x(s)$.

*Received by the editors September 25, 2001; accepted for publication (in revised form) by P. C. Hansen September 10, 2002; published electronically February 25, 2003. This work was supported in part by the National Science Foundation under grant CCR-97-32022 and by the Office of Naval Research under grant N000140110181.

http://www.siam.org/journals/simax/24-4/39544.html

†Applied Mathematics and Scientific Computation Program, University of Maryland, College Park, MD 20742. Current address: GAMS Development Corporation, 1217 Potomac Street NW, Washington, DC 20007 (armin@gams.com).

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

Since recording devices make only a finite number of measurements, the imaging model can be discretized and (1.1) can be written as a matrix equation. The discretized model is

$$Ax = b - r, \tag{1.2}$$

where the matrix A is the discretized counterpart of $a(s,t)$, and x and b also are the discretized versions of the corresponding continuous functions. If the blurring function a is assumed to be spatially invariant, then the matrix A has a special structure: for 1-dimensional signals it is Toeplitz and for 2-dimensional signals it is block Toeplitz with Toeplitz blocks.

If the cause of the blur, and hence a, is not known exactly, then our estimate of A has errors and the problem is known as *blind deconvolution*. In this case the model in (1.2) should be replaced by

$$(A + E)x = b - r, \tag{1.3}$$

a problem of the total least norm variety. If the matrix A has no special structure and the error $\|[E, r]\|_p$ is measured using the Frobenius norm, then the problem can be solved using the total least squares (TLS) method [5]. For image processing problems, the matrix A has a special structure, and it is desirable to enforce the same structure on the error matrix E. Rosen, Park, and Glick [23] developed the structured total least norm (STLN) method to solve such problems.

While STLN is useful for many structured linear problems, the blind deconvolution problem as encountered for image deblurring is generally ill-posed [9]. In particular, the matrix A is often ill-conditioned, resulting in poor recovered images when STLN is applied.

Regularization methods must be implemented in order to stabilize STLN and to obtain useful results. In this paper it is shown how to implement Tikhonov regularization [20, 26] to arrive at the regularized structured total least norm (RSTLN) algorithm. Implementation of Tikhonov regularization for constrained TLS problems had been developed previously [15, 17]. The first of these works predated the work of Rosen, Park, and Glick on the simpler problem. These works, however, focused solely on the 2-norm case. The contributions herein are the extension for $p = 1$ and $p = \infty$ norms and the comparison of methods. In the $p = 1$ and $p = \infty$ cases, the main computational task lies in solving a linear program (LP).

The paper is structured as follows. In the next section the STLN method is introduced and derived. In section 3 the general RSTLN method is introduced and derivations are presented for the $p = 1$, 2, and ∞ cases. Finally, in section 4 we present numerical results, and in section 5 we draw conclusions.

2. Derivation of the STLN method. In order to understand the RSTLN method, a brief derivation of STLN based on [23] is given. For a more thorough derivation, the reader is referred to [23] and [12].

2.1. TLS and STLN. The TLS [5] formulation for solving problems as in (1.3) is to find a matrix E and a vector r such that

$$\|[E,\ r]\|_F \tag{2.1}$$

is minimized, where F denotes the Frobenius norm and $r = b - (A + E)x$ is the residual. If the matrix A has a special structure which the user wants to enforce

on E, then the TLS formulation is not applicable. Instead, the STLN formulation proves useful.

As in [23], assume that the matrix $E \in \mathbf{R}^{m\times n}$ is parameterized by elements of the vector $\alpha \in \mathbf{R}^q$, $q < mn$. Then the residual is a function of α and x. The STLN formulation is to find vectors α and x such that

$$\left\| \begin{array}{c} r(\alpha, x) \\ D\alpha \end{array} \right\|_p \tag{2.2}$$

is minimized, where $p = 1$, 2, or ∞ and D is a diagonal *weighting matrix* through which the size of α is measured. Note that the norm in (2.2) is a norm over the space of structured matrices crossed with vectors in $\mathbf{R}^m$. For $p = 2$, it is the same as the Frobenius norm in (2.1) but, for $p = 1$ and $p = \infty$, it is not equivalent to any matrix norm.

If the elements of E are linear functions of the parameters α, then there exists a matrix X parameterized by x such that

$$X\alpha = Ex. \tag{2.3}$$

For a detailed description on construction of the matrix X, see [23] and [12]. Note that if the matrix E is structured, then so is X.

Now let Δx and ΔE denote small changes in x and E, respectively. Then

$$X\Delta\alpha = (\Delta E)x. \tag{2.4}$$

If we expand $r(\alpha, x)$ in a Taylor series about $[\alpha^T \; x^T]^T$ and ignore second order and higher terms, we have

$$\begin{array}{rcl} r(\alpha + \Delta\alpha, x + \Delta x) & \approx & b - (A + E)x - X\Delta\alpha - (A + E)\Delta x \\ & = & r(\alpha, x) - X\Delta\alpha - (A + E)\Delta x. \end{array} \tag{2.5}$$

Hence, we have a linearization of (2.2),

$$\min_{\Delta\alpha,\Delta x} \left\| \begin{bmatrix} X & A + E \\ D & 0 \end{bmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} + \begin{pmatrix} -r \\ D\alpha \end{pmatrix} \right\|_p. \tag{2.6}$$

The general idea behind the STLN method is to start with some initial estimates for x and E, solve the minimization problem in (2.6) for $\Delta\alpha$ and Δx, set $x = x + \Delta x$ and $\alpha = \alpha + \Delta\alpha$, and update the residual r and the matrices E and X. The procedure is repeated iteratively until $\|\Delta\alpha\|$ and $\|\Delta x\|$ are below a specified tolerance, at which point the algorithm has converged to a solution. For a detailed description the reader is referred to [23].

3. Derivation of RSTLN. In order to make STLN more robust in the presence of noise (as is encountered in most image deblurring applications), a form of regularization must be introduced. The method of Tikhonov [26] is implemented herein, which prevents the solution x from becoming too large. In particular, (2.2) can be modified to arrive at the RSTLN algorithm. The new problem formulation is to find vectors α and x so that

$$\left\| \begin{array}{c} r(\alpha, x) \\ D\alpha \\ \lambda x \end{array} \right\|_p \tag{3.1}$$

TABLE 3.1

RSTLN Algorithm

1. Set $E = 0_{m\times n}$ and $\alpha = 0_{q\times 1}$.
2. Compute x by $\min_x \|Ax - b\|_p$ (for $p = 2$ this is just least squares).
3. Compute X from x and the residual $r = b - Ax$.
4. For $k = 1, 2, \ldots$ until $\|\Delta x\|, \|\Delta\alpha\| \le \epsilon$ repeat steps 4.1–4.3.
   - 4.1. Solve
   $$\min_{\Delta\alpha,\Delta x} \left\| \begin{bmatrix} X & A+E \\ D & 0 \\ 0 & \lambda I \end{bmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} + \begin{pmatrix} -r(\alpha, x) \\ D\alpha \\ \lambda x \end{pmatrix} \right\|_p .$$
   - 4.2. Set $x = x + \Delta x$ and $\alpha = \alpha + \Delta\alpha$.
   - 4.3. Construct E from α, and X from x and compute $r = b - (A + E)x$.
5. The recovered image is x and the recovered blurring matrix $(A + E)$.

is minimized, where λ is a positive scalar known as the regularization parameter and $p = 1$, 2, or ∞. More generally, we could replace λx by λLx, where L is an operator chosen to force some desirable property on the solution x. For example, L might be a difference operator if we want a smooth image; see, for example [8, sect. 4.3]. For simplicity, we will write the algorithm for the case $L = I$, although the generalization is straightforward.

Using the relation in (2.5) and similar reasoning as for the STLN method, the linearization of (3.1) results in

$$\min_{\Delta\alpha,\Delta x} \left\| \begin{bmatrix} X & A+E \\ D & 0 \\ 0 & \lambda I \end{bmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} + \begin{pmatrix} -r \\ D\alpha \\ \lambda x \end{pmatrix} \right\|_p . \tag{3.2}$$

The general RSTLN algorithm (for arbitrary norm p) is listed in Table 3.1. We remark that Tikhonov regularization can be added in the same manner to the structured nonlinear total least norm (SNTLN) method [24], a variant of STLN for structured nonlinear parameter estimation problems. The resulting regularized algorithm is similar to RSTLN and may be the focus of future work.

3.1. RSTLN for $p = 2$. The minimization problem in the RSTLN formulation is equivalent to minimizing the function

$$\phi(\alpha, x) = \frac{1}{2}\|r(\alpha, x)\|_2^2 + \frac{1}{2}\|D\alpha\|_2^2 + \frac{1}{2}\|\lambda x\|_2^2. \tag{3.3}$$

The 2-norm case has the property of differentiability so that Gauss–Newton theory is applicable. Using similar reasoning as in [23] for the STLN method, it follows that step 4.1 is a Gauss–Newton method which approximates the Hessian of $\phi(\alpha, x)$ by the positive definite matrix $M^T M$, where

$$M = \begin{bmatrix} X & A+E \\ D & 0 \\ 0 & \lambda I \end{bmatrix}. \tag{3.4}$$

See also [3].

The least squares normal equations can be solved using the conjugate gradient method, where the Toeplitz (or block Toeplitz with Toeplitz blocks) structure of M is exploited. In particular, the FFT is used for efficient computation of matrix-vector products.

Another, more efficient, approach for $p = 2$ may be to apply the techniques of [14] for the nonregularized STLN to RSTLN. In particular, an algorithm based on the generalized Schur algorithm [16] for solving least squares problems is used which exploits the structure of the matrix

$$\begin{bmatrix} X & A+E \\ D & 0 \end{bmatrix}. \tag{3.5}$$

Since the RSTLN matrix M has a similar structure to this, the method in [14] should be applicable. This may be the focus of future work.

3.2. RSTLN for $p = \infty$. For both the $p = 1$ and $p = \infty$ cases, step 4.1 of the RSTLN algorithm is an LP. To see this, an approach similar to that in [23] is used.

Let us first consider the derivation for $p = \infty$. Suppose the original image in vector form is $x \in \mathbf{R}^{n\times 1}$, that $\alpha \in \mathbf{R}^{q\times 1}$, and that the residual $r \in \mathbf{R}^{m\times 1}$. Then the optimal function value in step 4.1 is $\bar{\sigma}$, where $\bar{\sigma}$ is determined from the LP

$$\begin{array}{lrcccl} \min\limits_{\Delta\alpha,\Delta x,\bar{\sigma}} & \bar{\sigma} & & & & \\ \text{subject to} & -\bar{\sigma}e_m & \le & X\Delta\alpha + (A+E)\Delta x - r & \le & \bar{\sigma}e_m, \\ & -\bar{\sigma}e_q & \le & D\Delta\alpha + D\alpha & \le & \bar{\sigma}e_q, \\ & -\bar{\sigma}e_n & \le & \lambda\Delta x + \lambda x & \le & \bar{\sigma}e_n, \end{array} \tag{3.6}$$

where $e_k \in \mathbf{R}^{k\times 1}$ is a vector of ones.

Using the matrix M we can write the linear programming problem in standard form,

$$\begin{array}{ll} \min\limits_{\Delta\alpha,\Delta x,\bar{\sigma}} & \bar{\sigma} \\ \text{subject to} & \begin{bmatrix} M & -e_{m+n+q} \\ -M & -e_{m+n+q} \end{bmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \\ \bar{\sigma} \end{pmatrix} \le \begin{pmatrix} r \\ -D\alpha \\ -\lambda x \\ -r \\ D\alpha \\ \lambda x \end{pmatrix}. \end{array} \tag{3.7}$$

Depending on the method used to solve the LP, it may be useful to consider the dual formulation. Setting $\sigma = -\bar{\sigma}$, it follows that the dual is

$$\begin{array}{ll} \min\limits_{y_i \ge 0} & r^T y_1 - \alpha^T D y_2 - \lambda x^T y_3 - r^T y_4 + \alpha^T D y_5 + \lambda x^T y_6 \\ \text{subject to} & \begin{bmatrix} M^T & -M^T \\ e_{m+n+q}^T & e_{m+n+q}^T \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \end{array} \tag{3.8}$$

where $y_1, y_4 \in \mathbf{R}^{m\times 1}$, $y_2, y_5 \in \mathbf{R}^{q\times 1}$, and $y_3, y_6 \in \mathbf{R}^{n\times 1}$. System (3.8) can be solved using any standard simplex or interior point method.

The reader should note that since the matrix M has a special structure (Toeplitz or block Toeplitz with Toeplitz blocks), any practical implementation of RSTLN for $p = 1$ or $p = \infty$ should exploit this property when solving the LP.

3.3. RSTLN for $p = 1$. The derivation for the $p = 1$ case is similar to the $p = \infty$ case. Again, let $\bar{\sigma}$ be the optimal function value in step 4.1. In particular, assuming x, α, and r are defined as previously, we have that $\bar{\sigma}$ is determined by

$$\begin{aligned} &\min_{\Delta\alpha,\Delta x,\bar{\sigma}} \quad \bar{\sigma} = \sum_{i=1}^{m} \bar{\sigma}_{1_i} + \sum_{i=1}^{q} \bar{\sigma}_{2_i} + \sum_{i=1}^{n} \bar{\sigma}_{3_i} \\ &\text{subject to} \quad \begin{array}{rcccl} -\bar{\sigma}_1 & \le & X\Delta\alpha + (A+E)\Delta x - r & \le & \bar{\sigma}_1, \\ -\bar{\sigma}_2 & \le & D\Delta\alpha + D\alpha & \le & \bar{\sigma}_2, \\ -\bar{\sigma}_3 & \le & \lambda\Delta x + \lambda x & \le & \bar{\sigma}_3, \end{array} \end{aligned} \tag{3.9}$$

where $\bar{\sigma}_1 \in \mathbf{R}^{m\times 1}$, $\bar{\sigma}_2 \in \mathbf{R}^{q\times 1}$, and $\bar{\sigma}_3 \in \mathbf{R}^{n\times 1}$. Using the matrix M we can write the LP as

$$\begin{aligned} &\min_{\Delta\alpha,\Delta x,\bar{\sigma}} \quad \bar{\sigma} = \sum_{i=1}^{m} \bar{\sigma}_{1_i} + \sum_{i=1}^{q} \bar{\sigma}_{2_i} + \sum_{i=1}^{n} \bar{\sigma}_{3_i} \\ &\text{subject to} \quad \begin{bmatrix} M & -I_{m+n+q} \\ -M & -I_{m+n+q} \end{bmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \\ \bar{\sigma}_1 \\ \bar{\sigma}_2 \\ \bar{\sigma}_3 \end{pmatrix} \le \begin{pmatrix} r \\ -D\alpha \\ -\lambda x \\ -r \\ D\alpha \\ \lambda x \end{pmatrix}. \end{aligned} \tag{3.10}$$

As for the $p = \infty$ case, the user may want to use the dual formulation. Setting $\sigma = -\bar{\sigma}$, our formulation becomes

$$\begin{aligned} &\min_{y_i \ge 0} \quad r^T y_1 - \alpha^T D y_2 - \lambda x^T y_3 - r^T y_4 + \alpha^T D y_5 + \lambda x^T y_6 \\ &\text{subject to} \quad \begin{bmatrix} M^T & -M^T \\ I_{m+n+q} & I_{m+n+q} \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} 0_{m\times 1} \\ 0_{q\times 1} \\ 0_{n\times 1} \\ e_m \\ e_q \\ e_n \end{pmatrix}, \end{aligned} \tag{3.11}$$

where all y_i are as defined previously for the ∞-norm case, and $0_{k\times 1}$ is a vector of zeros.

3.4. Convergence of RSTLN for $p = 1$ or $p = \infty$. As for the STLN problem, the function minimized in (3.1) is nonconvex, so that there is no guarantee that the RSTLN algorithm converges to a global minimum. For the $p = 2$ norm case the Gauss–Newton theory is applicable: a suitable line search method (see, for example, [3]) can be used to guarantee convergence to a local minimizer from any starting point.

For $p = 1$ and $p = \infty$, Gauss–Newton theory is not directly applicable since differentiability is lost. But the essential idea is the same as that for the $p = 2$ norm.

In particular, the solutions $\begin{bmatrix}\Delta\alpha^T & \Delta x^T\end{bmatrix}^T$ to the LPs given in (3.7) and (3.10) can be thought of as descent directions to the function in (3.1) for the respective p-norm. Again, in order to guarantee convergence to a local minimizer from any starting point, a line search method can be implemented.

4. Numerical results. In this section, experiments are presented to show that the RSTLN method deblurs images better than the STLN method. In particular, examples are shown comparing RSTLN and STLN for the $p = 1$, 2, and ∞ norms. We also compare our results with other blind deconvolution algorithms.

4.1. Experimental design.

4.1.1. Numerical issues. All of our code was written in MATLAB to take advantage of its image visualization capabilities.

We constructed our experiments by taking a known image x_{true} (stretched out to a vector by stacking the columns of the image) and a known blurring function A_{true} and using them to construct a blurred image $b_{true} = A_{true}x_{true}$. Then we added *n-bit noise* to the data: the elements of A_{noisy} were equal to those of A_{true} plus noise from a normal distribution with mean zero and standard deviation $\max_{i,j} A_{true}(i,j)/2^n$. Similarly, b_{noisy} was equal to $A_{noisy}x_{true}$ plus noise with standard deviation $\max_i b_{true}(i)/2^n$. Thus, the data perturbation can be measured by

$$\begin{aligned} pert(b) &= \|b_{noisy} - b_{true}\|_2/\|b_{true}\|_2, \\ pert(A) &= \|A_{noisy} - A_{true}\|_F/\|A_{true}\|_F. \end{aligned}$$

To evaluate the algorithms, we took the computed (recovered) image x_{rec} and the computed blurring function $(A+E)_{rec}$ and computed relative errors

$$\begin{aligned} err(x) &= \|x_{rec} - x_{true}\|_2/\|x_{true}\|_2, \\ err(A) &= \|(A+E)_{rec} - A_{true}\|_F/\|A_{true}\|_F, \\ err(b) &= \|b_{rec} - b_{true}\|_2/\|b_{true}\|_2, b_{rec} = (A+E)_{rec}x_{rec}. \end{aligned}$$

4.1.2. Implementation issues for STLN and RSTLN. For the STLN and RSTLN algorithms, a linear problem needs to be solved at each iteration; see step 4.1 of Table 3.1. For the $p = 2$ norm, we used the conjugate gradient least squares method to solve this problem. We set the conjugate gradient termination condition to a relative residual tolerance of 10^{-6} or 1000 iterations. This generally produces satisfactory accuracy to determine the descent direction, but for larger images the maximum number of iterations was sometimes taken.

For the $p = 1$ and $p = \infty$ cases we solved the LP in step 4.1 using the MATLAB function `linprog.m` with the *largescale* model employed. The function uses the LIPSOL [27] algorithm and is based on a primal-dual interior point method. Because of limitations in the MATLAB interface to LIPSOL, we were only able to set our stopping criteria to 10^{-2} to 10^{-3} compared with tolerances of 10^{-6} for the STLN experiments in [23]; a smaller tolerance caused LIPSOL to fail to converge. Even with this difficulty, RSTLN gives better results than STLN. Our current implementation is restricted to fairly small images because of the large number of constraints in the LP. While the constraint matrix M passed into `linprog.m` is sparse, its factorization generally is not. Hence, the LP solver as implemented in MATLAB is very memory intensive and currently restricts our test cases to images no larger than 100×100.

We stop the STLN or RSTLN iterations when the relative change in the recovered image and the recovered A matrix drops below some tolerance *tol*. At times

we stopped the RSTLN method prematurely before reaching the desired tolerance because for a higher number of iterations the reconstructed image was, in fact, deteriorating. This is a common phenomenon in the numerical solution of ill-posed problems shared, for example, by the Lucy–Richardson (LR) algorithm, and the number of iterations can be viewed as an additional regularization parameter [8, Chap. 6], [10]. Initial iterations tend to reconstruct the image while later ones tend to focus on the noise. Problems with low signal-to-noise ratios are particularly prone to such noise amplification; the basic problem is that we do not want to minimize the function in (2.1) but just drive its value down to noise level. Thus, in our experiments, a lower number of RSTLN iterations sometimes yielded a better recovered image than one recovered using more iterations, even if the latter yielded a better function value for (3.1) and satisfied lower tolerances.

The choice of the regularization parameter λ for algorithms such as RSTLN is a well-studied problem (see, for example, [4, 7, 18, 19] and [8, Chap. 7]). Ideally, the choice balances the need to stay close to the original noise-contaminated problem without causing its ill-conditioning to produce unacceptable noise in the solution. In our experiments, we were concerned with the best solution obtainable for any choice of parameter. We set $D = I$ and solved each problem for a wide range of values $\lambda > 0$, choosing the parameter resulting in the smallest value for the 2-norm of the image error. The solution was sometimes quite sensitive to this choice.

4.1.3. Comparison with other blind deconvolution methods. We compare RSTLN with two other blind deconvolution methods: the blind LR method and the APEX/SECB method of Carasso.

The blind LR algorithm is an extension of the well-known original LR method [13, 22] to problems in which the blurring function is unknown. The original iterative method was derived from Bayes' theorem and assumes that the blurred image, the original image, and the point spread function (PSF) are (possibly nonnormalized) probability density functions. The most common and efficient implementation makes use of the FFT to compute convolutions. This implicitly imposes periodic boundary conditions on the image.

The blind version is similar to the original method; each iteration alternately uses several iterations of the nonblind algorithm to estimate a new PSF and then a new image. It is generally more effective for images having many pixels and for images with fewer sharp edges, since convolution tends to smooth edge boundaries [9].

The algorithm can be used without FFTs, but it is computationally much slower and may produce *ringing* (high frequency oscillations) in the recovered image if the image does not have finite support. This ringing arises because the method has a probabilistic basis, and any implementation must conserve energy. Thus, a nonperiodic (for example, zero boundary condition) implementation is useful only for images having support strictly inside the image boundaries. Convolutions involving images that violate this assumption do not conserve energy since data outside of the original image boundary are lost; this lost energy tends to be recovered as ringing. Conservation of energy, image support, and ringing are discussed in more detail in [21]. To reduce the amount of ringing, we experimented with techniques such as *tapering*, implemented in the MATLAB routine `edgetaper.m`, which seek to transform a nonperiodic image to a periodic one by reblurring the edges of the image with a suitable PSF. The reader is referred to [25] for details.

The stopping criterion for MATLAB's blind LR function `deconvblind.m` is based solely on the input number of iterations. The user may specify this total number of

TABLE 4.1

RSTLN errors for $p = 1$, 2, and ∞. We list the errors in the image x, the matrix A, and the residual error $err(b)$ for the unregularized STLN and the RSTLN methods for each of the norms. For the $p = 1$ and $p = 2$ norms the RSTLN recovered image error $err(x)$ is much smaller than for STLN. For $p = \infty$ the image error is near optimal and the error using RSTLN is only slightly smaller than for STLN.

Test Case 1	$err(x)$	$err(A)$	$err(b)$
$p = 2$ STLN	1.19	3.97e−2	1.1e−3
$p = 2$ RSTLN	0.39	4.10e−2	1.1e−3
$p = 1$ STLN	0.97	3.99e−2	1.4e−3
$p = 1$ RSTLN	0.44	4.00e−2	1.1e−2
$p = \infty$ STLN	0.50	4.02e−2	5.5e−1
$p = \infty$ RSTLN	0.45	3.98e−2	4.9e−1

iterations or use the default value of 10. Our non-FFT implementation is similar to the nonblind MATLAB routine `deconvlucy.m` but lets the user specify the total number of iterations and, for each, the number of LR inner iterations to update the image and PSF estimates. We estimate the optimal number of iterations by recovering images using a wide variety of choices and then choosing the image resulting in the smallest 2-norm error. For our comparison test cases, where our goal was to show only general trends in the recovered images, we often used a default of 10 iterations, modifying this number as needed.

Carasso's APEX/SECB method [1] can be applied to the class of PSFs a whose FFT, denoted by $\hat{a}(\xi, \eta)$, is of the form

$$\hat{a}(\xi, \eta) = e^{-\alpha(\xi^2+\eta^2)^\beta}, \tag{4.1}$$

where ξ and η are the respective frequency coordinates. If the blurred image $b = a \otimes x$ is obtained by (periodic) convolution, then in the Fourier domain,

$$\begin{aligned} \hat{b}(\xi, \eta) &= \hat{x}(\xi, \eta) \cdot \hat{a}(\xi, \eta) \\ &= \hat{x}(\xi, \eta) \cdot e^{-\alpha(\xi^2+\eta^2)^\beta}. \end{aligned} \tag{4.2}$$

The idea behind the PSF identification method is to fit the function $\alpha|\xi|^{2\beta}$ to the logarithm of the Fourier transform of the blurred image minus an estimate of the true image; see [1] for details. If the image or the PSF fails to meet necessary requirements, then such a fit will not be possible.

4.2. Results.

Test 1. Our first test consists of a cross of size 21×21. The true PSF is a Gaussian blur with variance 2.5, truncated to a support of size 11×11.

The blurred image was obtained by convolving the original image and PSF, assuming that pixel values outside the image are zero (zero boundary conditions). The original and blurred images are shown in Figures 4.1(A) and (B). 6-bit noise was added to the PSF to obtain the initial PSF estimate. This resulted in $pert(A) = 3.99 \times 10^{-2}$. Furthermore, 11-bit noise was added to the blurred image, resulting in $pert(b) = 1.10 \times 10^{-3}$.

The errors resulting from the STLN and RSTLN methods for the different p-norms are shown in Table 4.1. The corresponding images are shown in Figures 4.1(C)–(H). From the error table we see that the use of RSTLN generally increases the error $err(A)$ in the blurring matrix and the residual error $err(b)$. For the 1- and 2-norms, however,

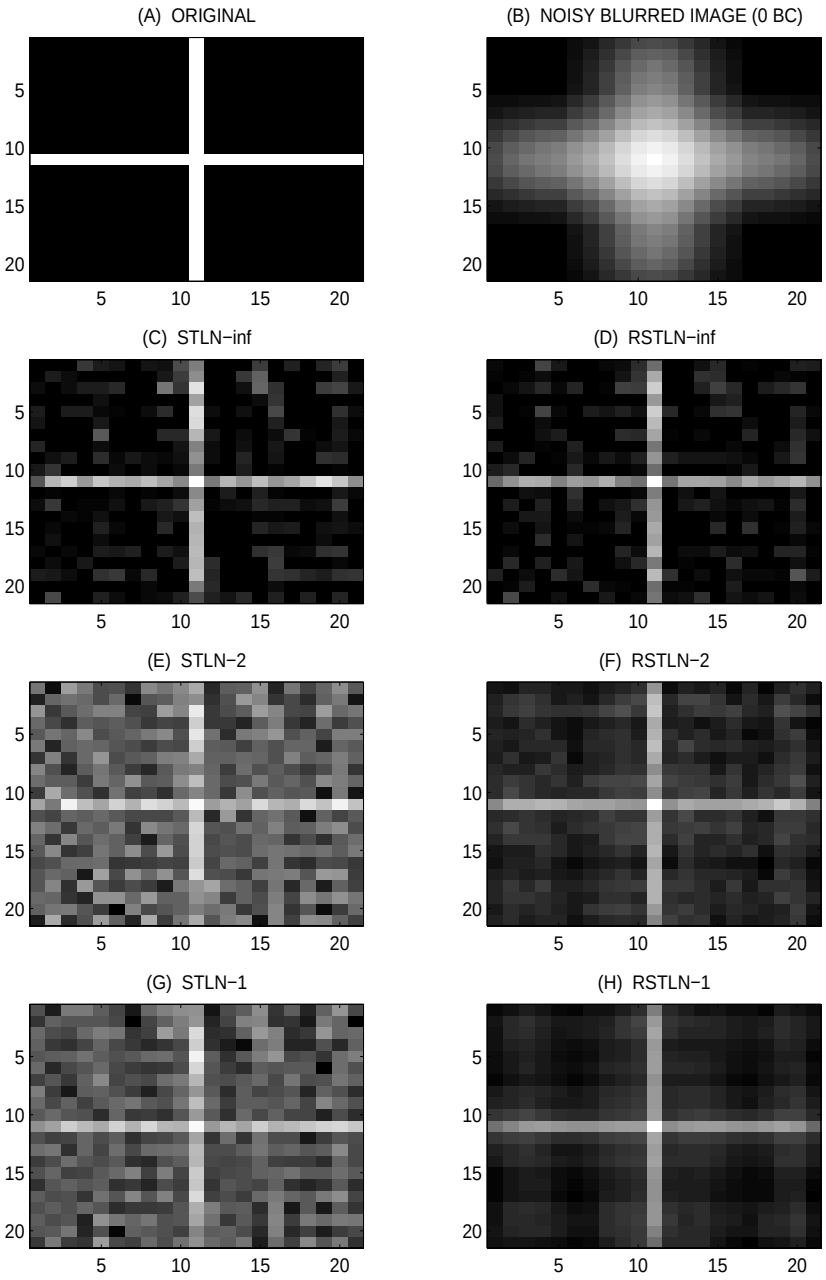

FIG. 4.1. *RSTLN cross (noise, Gaussian blur). Test* 1*, results of STLN and RSTLN methods using* $p = 1$, 2, *and* ∞ *norms. Random noise is present in the blurred image. The blur estimate is the true blur plus the addition of* 6*-bit noise so that* $pert(A) = 3.99 \times 10^{-2}$. 11*-bit noise was added to the blurred image so that* $pert(b) = 1.10 \times 10^{-3}$. (A) *Original image,* 21×21. (B) *Noisy blurred image (zero BC).* (C) *STLN* (∞*-norm) solution with* $tol = 10^{-2}$. *Solution is near optimal:* 13 *iterations.* (D) *RSTLN* (∞*-norm) recovered image with* $tol = 10^{-2}$, *regularization parameter* $\lambda = 0.001$, 12 *iterations.* (E) *STLN* (2*-norm) solution with* $tol = 10^{-3}$, 22 *iterations.* (F) *RSTLN* (2*-norm) recovered image with* $tol = 10^{-3}$, $\lambda = 0.05$, 27 *iterations.* (G) *STLN* (1*-norm) solution with* $tol = 10^{-2}$, 13 *iterations.* (H) *RSTLN* (1*-norm) recovered image with* $tol = 10^{-2}$, $\lambda = 0.5$, 50 *iterations.*

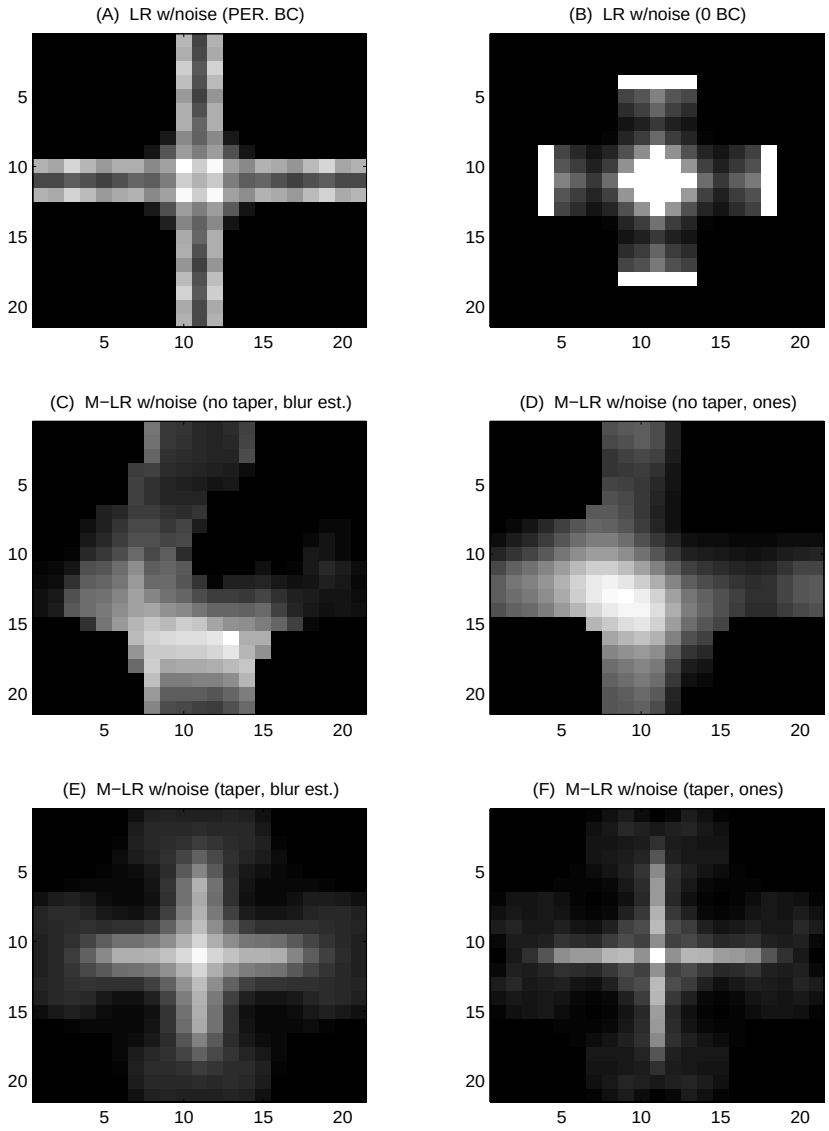

FIG. 4.2. *Test* 1, *LR results.* (A) *Periodic LR implementation using a periodic blurred image,* 20 *LR iterations each with* 10 *iterations.* (B) *Zero boundary condition LR implementation using a zero BC blurred image,* 5 *LR iterations each with* 10 *iterations.* (C) *M-LR result without tapering and using the RSTLN initial PSF estimate,* 10 *iterations.* (D) *M-LR result without tapering and using an* 11×11 *matrix of ones for the initial PSF estimate,* 10 *iterations.* (E) *M-LR result with tapering and using the RSTLN initial PSF estimate,* 50 *iterations.* (F) *M-LR result with tapering and using an* 11×11 *matrix of ones for the initial PSF estimate,* 100 *iterations.*

the error $err(x)$ in the image estimate is considerably lower, so the reconstructed image is improved. For the $p = \infty$ norm, the image obtained from STLN was near optimal, and all RSTLN experiments for nonzero values of the regularization parameter λ resulted in higher image errors.

In Figure 4.2 we present the results of the blind LR method. In (A) we show results obtained by LR in reconstructing images blurred with periodic boundary conditions

TABLE 4.2

RSTLN errors for $p = 2$ for the large cross test case. We list the errors in the image x, the matrix A, and the residual error $err(b)$ for the unregularized STLN and the RSTLN methods for $p = 2$. For the RSTLN ($\lambda = 2.5$) recovered image error $err(x)$ is much smaller than for STLN.

Test Case 2	$err(x)$	$err(A)$	$err(b)$
$p = 2$ STLN	4.2895	4.03e−2	1.03e−2
$p = 2$ RSTLN	0.5885	1.15e+0	9.20e−3

(6-bit noise added) using 20 outer iterations with 10 LR iterations in each. The width of the cross is broadened due to blurring of the edges during the reconstruction.

In Figures 4.2(B)–(F), we present the result of various attempts to reconstruct the image with zero boundary conditions from Figure 4.1. In (B) we show the result obtained by using 5 outer iterations with 10 LR iterations each, computing convolutions using zero padded images. It is clear that the image is distorted, and ringing is observed throughout. The other images are reconstructed using the MATLAB-supplied implementation of blind LR, which we call M-LR. In (C) we show the M-LR result, beginning with the blur estimate as for RSTLN, and stopping after the MATLAB default of 10 iterations. We repeat this experiment in (D) but starting from a flat PSF estimate (a matrix of ones of size 11×11). In both cases only poor reconstructions are obtained. In (E) and (F) we show similar results as in (C) and (D), except that the image is tapered using `edgetaper.m`. The reader is referred to [25] for details. We performed 50 and 100 M-LR iterations, respectively. The reader should note that the algorithm is not able to reconstruct data near the image boundary, although the interior is adequately recovered.

The APEX/SECB method cannot be applied to this image because it is too small to yield enough data points.

Test 2. Our next test consists of a somewhat broader cross image of size 41×41 with a nonzero cross width of 5. The image was blurred with an 11×11 Gaussian. 8-bit noise was added to the blurred images, resulting in $pert(b) = 1.05 \times 10^{-2}$ and 9.8×10^{-3}, respectively. The blur estimate was obtained by adding 6-bit noise to the original blur, resulting in $pert(A) = 3.91 \times 10^{-2}$.

Again, we present results comparing the STLN, RSTLN, LR, and M-LR methods, as well as Carasso's APEX/SECB method. In Figures 4.3(A) and (B) we show the original and blurred images. In (C) we show the STLN 2-norm solution (that is, without any regularization), and in (D) the best RSTLN 2-norm solution with regularization (using $\lambda = 0.75$). (The RSTLN $p = 1$ and $p = \infty$ were not computed due to the expense of solving the linear programming problems.) The resulting STLN and RSTLN errors for the 2-norm are shown in Table 4.2.

For APEX/SECB, the original image in Figure 4.3(A) was blurred using periodic boundary conditions as in (4.2) using parameters $\alpha = 0.075$ and $\beta = 1$. This resulted in a blurred image nearly identical to (B). Again, 8-bit noise was added to the blurred image. In subplot (E) we show the results of using APEX/SECB for PSF identification and subsequent deblurring of the periodic noisy blurred image. The APEX PSF identification procedure resulted in parameter estimates of $\alpha_{est} = 0.0749$ and $\beta_{est} = 0.9756$, which are fairly close to the true parameter values. Unfortunately, this method was unsuccessful for images blurred with zero boundary conditions and noise added. In (F), we show the APEX optimization function for different scalar value image estimates. The nonsmooth family of curves corresponds to the optimization function for different scalar estimates for the unknown frequency-domain

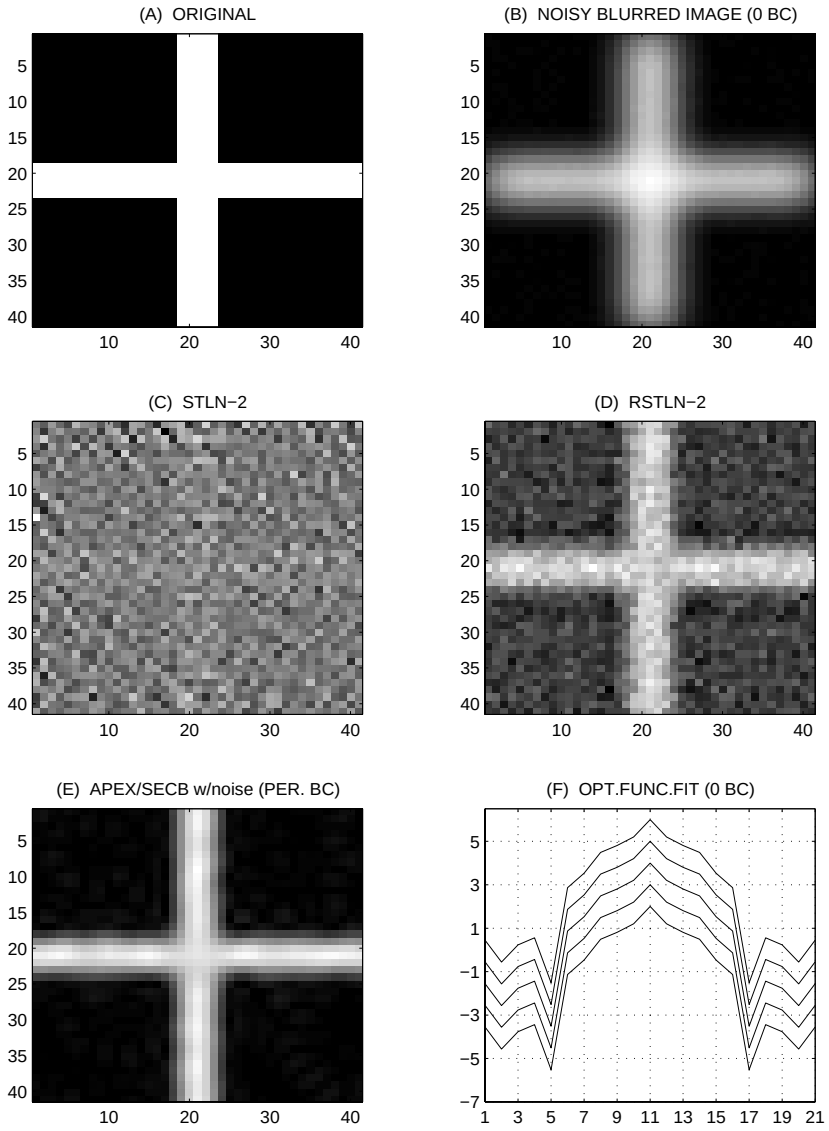

FIG. 4.3. *Test* 2*, RSTLN, and APEX/SECB results. The image was blurred using zero boundary conditions.* 8*-bit noise was added to obtain the image in* (B)*, resulting in* $pert(b) = 1.05 \times 10^{-2}$*. The blur estimate was obtained by adding* 6*-bit noise to the original blur, resulting in* $pert(A) = 3.91 \times 10^{-2}$*.* (C) *STLN* 2*-norm solution,* $tol = 10^{-3}$*,* 26 *iterations.* (D) *Best RSTLN* 2*-norm solution,* $\lambda = 0.75$*,* $tol = 10^{-3}$*,* 25 *iterations.* (E) *APEX/SECB recovered image using a noisy periodic image. The image was blurred as in* (4.2) *using parameters* $\alpha = 0.075$ *and* $\beta = 1$*. The recovered PSF parameter estimates are* $\alpha_{est} = 0.0749$ *and* $\beta_{est} = 09756$ *using a scalar image component estimate of* $K = 2.2$*.* (F) *APEX optimization function for a zero BC noisy image. Since the function does not have the proper form* $\alpha|\xi|^{2\beta}$*, no fit can be obtained. In this case no PSF was found.*

image quantity $\log|\hat{x}^*(\xi, 0)|$ if the natural logarithm is applied to the right- and left-hand sides in (4.2) and when a noisy zero boundary condition blurred image is used. The curves do not have the proper form and thus do not permit a curve fit of the form $\alpha|\xi|^{2\beta}$. For this case no proper PSF can be found.

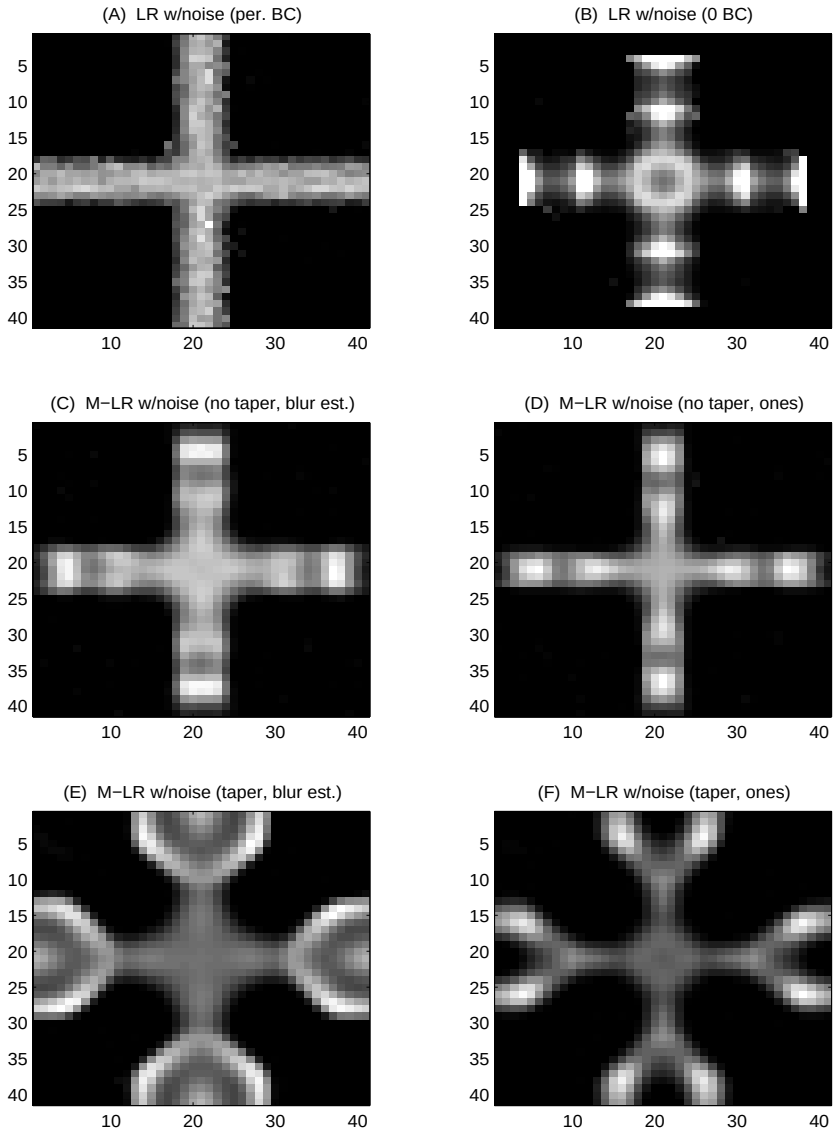

FIG. 4.4. *Test* 2*, LR results.* (A) *Periodic LR implementation using a periodic blurred image,* 50 *LR iterations each with* 10 *iterations.* (B) *Zero boundary LR implementation using a zero BC blurred image,* 50 *LR iterations each with* 10 *iterations.* (C) *M-LR result without tapering and using the RSTLN blur estimate,* 25 *iterations.* (D) *M-LR result without tapering and using an* 11×11 *matrix of ones for the PSF estimate,* 25 *iterations.* (E) *M-LR result with tapering and using the RSTLN blur estimate,* 10 *iterations.* (F) *M-LR result with tapering and using an* 11×11 *matrix of ones for the PSF estimate,* 10 *iterations.*

In Figure 4.4 we present results of the blind LR algorithm. In (A) we see that the algorithm gives a good result for periodic blurs, but the reconstruction for a zero boundary condition exhibits ringing and distortion. These results used 50 outer iterations, each using 10 LR iterations. In (B) we give the result for the zero boundary condition image using the zero boundary implementation. We then apply the M-LR algorithm to a noisy zero boundary blurred image. In (C) and (D) we show results

TABLE 4.3

RSTLN errors for $p = 2$ for the sun test case. We list the errors in the image x, the matrix A, and the residual error $err(b)$ for the unregularized STLN and the RSTLN methods for $p = 2$. For the RSTLN ($\lambda = 75$) recovered image error $err(x)$ is much smaller than for STLN.

Test Case 3	$err(x)$	$err(A)$	$err(b)$
$p = 2$ STLN	20.01	2.47e−2	2.19e−2
$p = 2$ RSTLN	0.9265	3.8483e+0	6.71e−1

using no tapering, 25 iterations, and an initial guess of either the RSTLN blur estimate or a matrix of ones of size 11×11. Both results exhibit ringing due to improper boundary conditions. In (E) and (F) we show M-LR results with tapering, using 10 outer iterations and initial blur estimates as in (C) and (D). The reconstructions are not useful.

Test 3. Our final comparison test consists of an image obtained from the NASA Image Exchange (http://nix.nasa.gov). It shows the corona of the sun and a large solar eruption. We truncated the image to size 99×99 and reduced it to grayscale.

Again, the image was blurred with a Gaussian PSF of size 11×11 in two ways: one assuming zero values for pixels outside the image, and the other assuming a periodic image. 6-bit noise was added to the image after blurring using a zero boundary condition. This resulted in $pert(b) = 2.20 \times 10^{-2}$. For the periodic image no noise was added to the blurred image. The blur estimate was obtained by adding 6-bit noise to the original blur ($pert(A) = 2.46 \times 10^{-2}$).

In Figure 4.5(A) we show the original and in (B) the noisy blurred image using zero boundary conditions. In (C) we show the STLN result using the 2-norm. Due to the high noise level in both the blurred image and the blur estimate, no useful result was obtained. In (D) we show the best result using the RSTLN method with a regularization value of $\lambda = 75$. We remark that in this case the algorithm did not converge to a tolerance of 10^{-2}. Instead, we stopped prematurely after 10 iterations. A larger number of iterations which did achieve the desired tolerance produced an image of lesser quality (see section 4.1.1 on noise amplification).

In Table 4.3 we computed the resulting errors for the STLN and RSTLN methods. Although $err(A)$ and $err(b)$ are increased for RSTLN with respect to STLN, clearly the image error is drastically reduced using the RSTLN method.

For the APEX/SECB method the image was blurred with a Gaussian blur using periodic boundary conditions and parameters $\alpha = 0.01$ and $\beta = 1$ as in (4.2). This resulted in a blurred image very similar to the one in Figure 4.5(B). 6-bit noise was added to the blurred image. Using the APEX PSF identification method, a curve fit to the optimization function was done, resulting in parameter estimates of $\alpha_{est} = 0.0108$ and $\beta_{est} = 1.028$. These are fairly close to the true PSF parameters. In (E) we show the APEX/SECB recovered image using the noisy blurred image with periodic boundary conditions. In (F) we show the function to be fit using the noisy image with zero boundary conditions. We plot the function using different scalar estimates for the original image component in (4.2). None of the functions have the proper form and a suitable curve fit of the form $\alpha|\xi|^{2\beta}$ is not possible. For this case no useful PSF was found.

In Figure 4.6 we show the results from the various LR experiments. In subplot (A) we have the LR result using a periodic image using our own periodic LR implementation. We performed 10 iterations, each with 10 LR iterations. In (B) we show the result using the zero boundary implementation and a zero boundary blurred

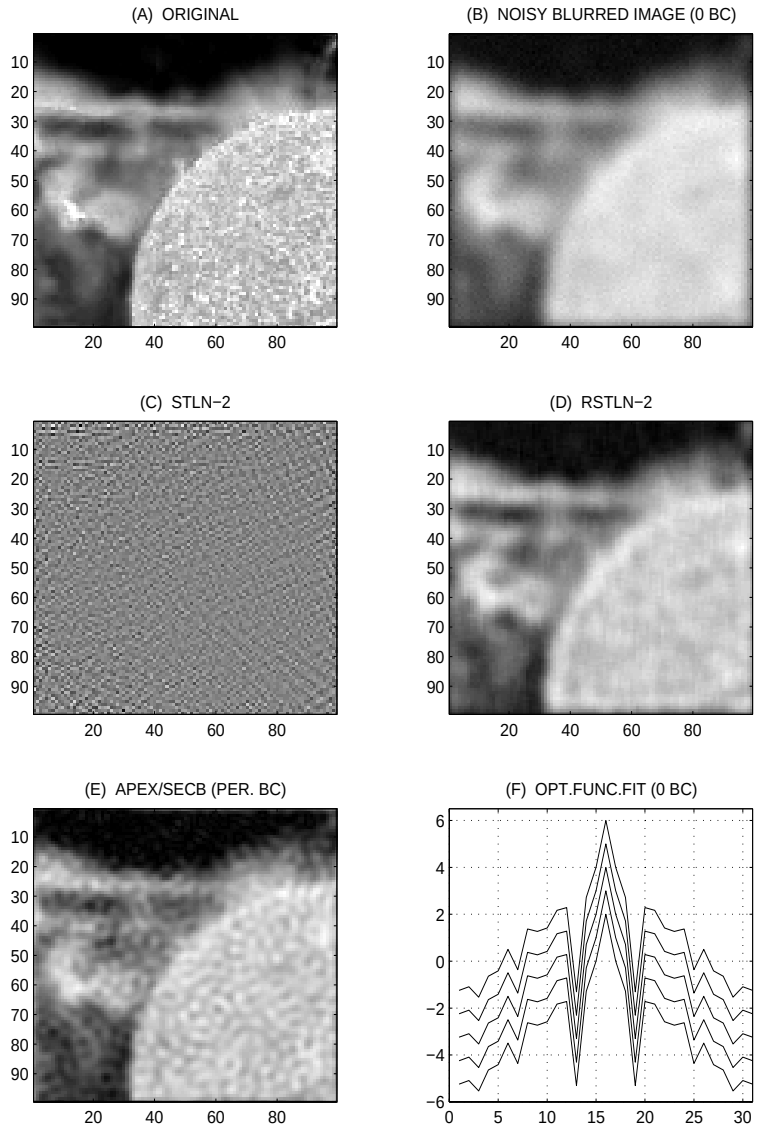

FIG. 4.5. *Test* 3, *RSTLN and APEX/SECB results.* (A) *Original image,* 99×99. (B) *Noisy blurred image (zero BC).* (C) *STLN (2-norm) solution with tol* $= 10^{-2}$, 2 *iterations.* (D) *RSTLN (2-norm) recovered image with initial tol* $= 10^{-2}$ *and regularization* $\lambda = 75$. *The experiment was stopped prematurely after* 10 *iterations. While a larger number of iterations did achieve the desired tolerance, the results were distorted by ringing.* (E) *APEX/SECB recovered image. Image is blurred assuming a periodic image as in* (4.2) *with parameters* $\alpha = 0.01$ *and* $\beta = 1$. (F) *Plot of optimization function if the image is blurred using zero BC. The different plots represent the optimization function for different scalar estimates for the unknown quantity* $\log|\hat{x}^*(\xi, 0)|$, *where* $\hat{x}^*(\xi, \eta)$ *denotes the normalized FFT of the original image* x. *Since none of the curves possess the proper shape, no useful PSF can be found.*

image. We performed 15 outer iterations, each with 10 iterations to estimate the new PSF and image. Severe ringing is present. In (C) and (D) we show the nontapered M-LR results using the RSTLN blur estimate, an 11×11 matrix of ones for the blur

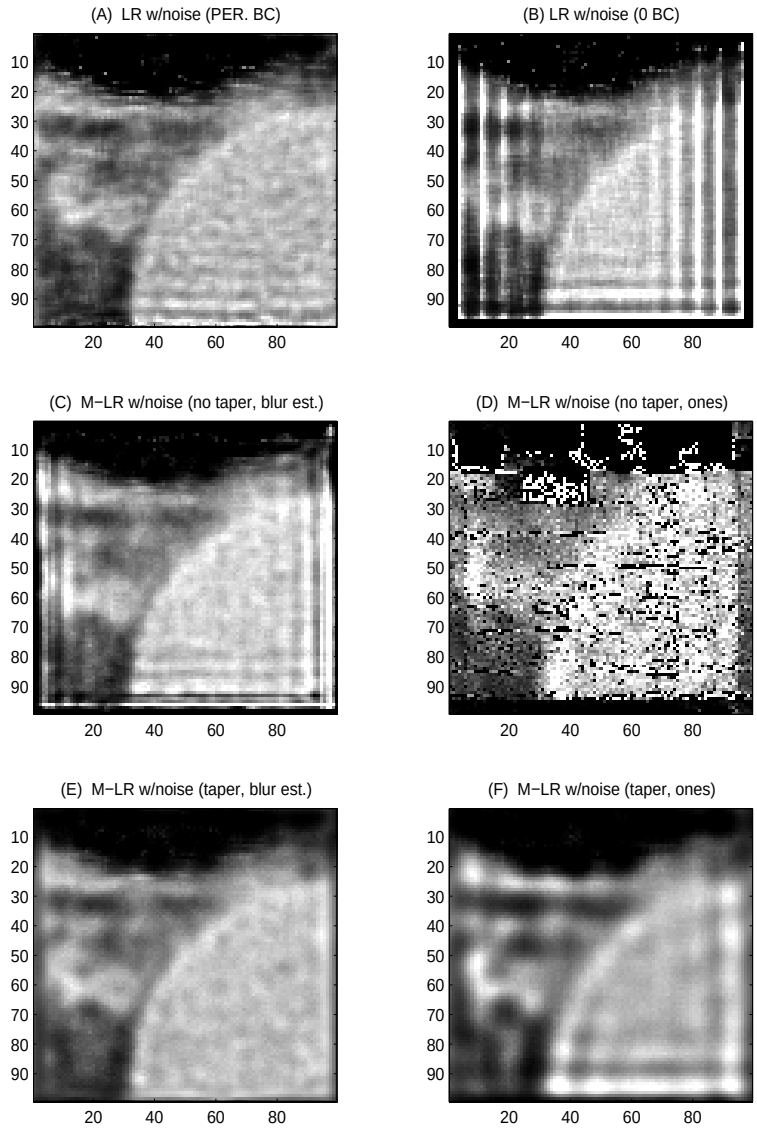

FIG. 4.6. *Test* 3*, LR results.* (A) *Periodic LR implementation using a periodic blurred image,* 10 *LR iterations each with* 10 *iterations.* (B) *Zero boundary LR implementation using a zero BC blurred image,* 15 *LR iterations each with* 10 *iterations.* (C) *M-LR result without tapering and using the RSTLN blur estimate,* 25 *iterations.* (D) *M-LR result without tapering and using an* 11×11 *matrix of ones for the PSF estimate,* 10 *iterations.* (E) *M-LR result with tapering and using the RSTLN blur estimate,* 25 *iterations.* (F) *M-LR result with tapering and using an* 11×11 *matrix of ones for the PSF estimate,* 10 *iterations.*

estimate, and a zero boundary blurred image. 25 outer iterations were performed, with 10 iterations each. For the result in (C), ringing is observed near the image boundary, whereas in (D) the image is severely distorted. Finally, in (E) and (F) we obtained results using M-LR and a tapered noisy blurred image using the two different initial blur estimate types. For the result in (E), 25 iterations were performed which produced reasonable results. The result in (F) was obtained after 10 iterations with less favorable results.

TABLE 4.4
Summary of methods.

Algorithm	Requirements	Comments
Blind LR	• FFT version requires periodic boundary conditions or finite support	• more effective with larger images • ringing if image fails to have finite support
Carasso's method	• periodic boundary conditions or finite support • image must have specific properties [1] • PSF must have specific properties [1]	• more effective with larger images
STLN	• substantial computation for 1- and ∞-norm methods	• sensitive to noise
RSTLN	• substantial computation for 1- and ∞-norm methods	• robust to noise

4.3. Effectiveness of methods. As the experiments indicate, some of the methods presented prove useful only if specific requirements are satisfied. This section summarizes the effectiveness of each of the methods.

The blind LR method using FFTs is useful only if the original image either was blurred using periodic boundary conditions or has finite support. If it does not satisfy either of these conditions, the recovered image often suffers from ringing. It is also observed that the method is sometimes more useful for larger images or if preprocessing techniques such as tapering or flat PSF initial estimates are used (see [25] for details).

Like the blind LR method, Carasso's APEX/SECB method requires periodic boundary conditions or finite support [2]. Furthermore, it can be applied only to the class of PSFs satisfying (4.1) and requires images to belong to a specific class as defined in [1].

In contrast, neither STLN nor RSTLN imposes any restrictions on the image or PSF and both are effective on small images. While STLN is useful for some total least norm problems, the blind deconvolution problem is generally ill-posed, so that small perturbations in the data can cause large changes in the solution. Thus, the RSTLN method proves to be more useful for most blind image deblurring applications where regularization is usually necessary.

If the noise is Gaussian, then least squares theory provides ample justification for choosing the 2-norm in RSTLN rather than the 1-norm or ∞-norm. However, in order to take advantage of this theory, the standard deviations of the two error distributions must be known so that the error terms can be balanced. When this data is unavailable, or when the noise distributions are not Gaussian, then the 1-norm and ∞-norm have no theoretical disadvantages. Our experiments show that the 1-norm in particular provides high-quality reconstructions and is not sensitive to outliers in the data.

A summary of the requirements and effectiveness of each method is given in Table 4.4.

5. Conclusions. We have presented the RSTLN algorithm for blind deconvolution. Like the STLN method, RSTLN preserves any affine structure in the matrix, and the user has the choice of minimizing the error for the 2-norm or for other norms such as the 1- and ∞-norms. The use of norms other than the 2-norm leads to good image recovery, although the cost is substantially higher.

In contrast to other methods, such as that of Carasso's APEX/SECB, the RSTLN method does not depend on having a periodic image. Ringing in the reconstructed images is less of a problem. Therefore, we can apply the RSTLN method for arbitrary boundary conditions, for example, zero (Dirichlet), Neumann (data outside the image boundary is a reflection of the corresponding data inside), or periodic.

Acknowledgments. The authors wish to thank Cleve Moler and Bruce Golden for their help with linear programming and MATLAB, Jon McCoy for spirited discussions on the LR method, and the referees for their insightful comments.

REFERENCES

[1] A. S. Carasso, *Direct blind deconvolution*, SIAM J. Appl. Math., 61 (2001), pp. 1980–2007.
[2] P. J. Davis, *Circulant Matrices*, Wiley, New York, 1979.
[3] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., Vol. 2, Wiley and Sons, New York, 1980.
[4] G. Golub, M. Heath, and G. Wahba, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics, 21 (1979), pp. 215–223.
[5] G. H. Golub and C. F. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.
[6] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1977.
[7] P. C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., 34 (1992), pp. 561–580.
[8] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1998.
[9] P. C. Hansen, *Numerical Aspects of Deconvolution*, Lecture Notes, Department of Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark, 2000.
[10] M. E. Kilmer and D. P. O'Leary, *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1204–1221.
[11] R. L. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images*, Kluwer Academic Publishers, Norwell, MA, 1991.
[12] P. Lemmerling, N. Mastronardi, and S. Van Huffel, *Fast algorithm for solving Hankel/Toeplitz structured total least squares problem*, Numer. Algorithms, 21 (2000), pp. 371–392.
[13] L. B. Lucy, *An iterative technique for the rectification of observed distributions*, Astronomical Journal, 79 (1974), pp. 745–754.
[14] N. Mastronardi, P. Lemmerling, and S. Van Huffel, *Fast structured total least squares algorithm for solving the basic deconvolution problem*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 533-553.
[15] V. Mesarović, N. Galatsanos, and A. Katsaggelos, *Regularized constrained total least squares image restoration*, IEEE Trans. Image Process., 4 (1995), pp. 1096–1108.
[16] B. De Moor, *Total least squares for affinely structured matrices and the noisy realization problem*, IEEE Trans. Signal Process., 42 (1994), pp. 3004–3113.
[17] M. K. Ng, R. J. Plemmons, and F. Pimentel, *A new approach to constrained total least squares image restoration*, Linear Algebra Appl., 316 (2000), pp. 237–258.
[18] D. P. O'Leary, *Near-optimal parameters for Tikhonov and other regularization methods*, SIAM J. Sci. Comput., 23 (2001), pp. 1161–1171.
[19] D. P. O'Leary and J. A. Simmons, *A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems*, SIAM J. Sci. Stat. Comput., 2 (1981), pp. 474–489.
[20] 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.
[21] A. Pruessner, *Blind Deconvolution Using a Regularized Structured Total Least Norm Algorithm*, Master's Thesis, University of Maryland, College Park, MD, 2001.
[22] W. H. Richardson, *Bayesian-based iterative method of image restoration*, J. Opt. Soc. Amer. A, 62 (1972), pp. 55–59.
[23] J. B. Rosen, H. Park, and J. Glick, *Total least norm formulation and solution for structured problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 110–126.
[24] J. B. Rosen, H. Park, and J. Glick, *Structured total least norm for nonlinear problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 14–30.

[25] J. Skilling and S. F. Gull, *Algorithms and applications*, in Proc. 1st Workshop on Maximum Entropy and Bayesian Methods in Inverse Problems, C. R. Smith and W. T. Grandy, Jr., eds., D. Reidel Publishing Company, Boston, 1985, pp. 83–132.

[26] A. N. Tikhonov, *Solution of incorrectly formulated problems and the regularization method*, Soviet Math. Dokl., 4 (1963), pp. 1035–1038.

[27] Y. Zhang, *Solving Large-Scale Linear Programs by Interior-Point Methods under the MATLAB Environment*, Technical Report, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1997.