ALGORITHMS FOR STRUCTURED TOTAL LEAST SQUARES PROBLEMS WITH APPLICATIONS TO BLIND IMAGE DEBLURRING *

ANOOP KALSI† AND DIANNE P. O'LEARY‡

Abstract. Mastronardi, Lemmerling, and van Huffel presented an algorithm for solving a total least squares problem when the matrix and its perturbations are Toeplitz. A Toeplitz matrix is a special kind of matrix with small displacement rank. Here we generalize the fast algorithm to any matrix with small displacement rank. In particular, we show how to efficiently construct the generators whenever M has small displacement rank and show that in many important cases the Cholesky factorization of the matrix M^TM can also be determined fast. We further extend this problem to Tikhonov regularization of ill-posed problems and illustrate the use of the algorithm on an image deblurring problem.

Key words. Displacement rank, block Toeplitz matrix, total least squares, structured total least squares, errors in variables method, image deblurring, Tikhonov regularization.

Running Title: Fast Structured Total Least Squares

1. Introduction.

In [4], Mastronardi, Lemmerling, and Van Huffel present an algorithm for solving fast structured total least squares problems of the form

$$\min_{E,\beta,x} \left\| \begin{bmatrix} E & \beta \end{bmatrix} \right\|_F^2 \tag{1.1}$$

subject to the constraints

$$(A+E)x = y + \beta$$

with $A \in \mathcal{R}^{m\times n}$ a given Toeplitz matrix and $y \in \mathcal{R}^{m\times 1}$ a given vector. They include one additional constraint: E is a Toeplitz matrix. They produced a fast algorithm for solving this *structured total least squares problem* (STLS) and showed that the solution was a better estimator than the solution to the total least squares problem without the Toeplitz constraint.

In this work, we consider the same problem (1.1), but under the constraint that A and E have small displacement rank relative to some matrices Z_1 and Z_2. Choosing these two matrices to be shift-down matrices and the rank to be two gives the Toeplitz constraint considered by [4], but we will be interested in other cases as well.

We also consider fast solution of the problem under the additional constraint that the norm of the solution vector x is specified. Note that this problem was posed in Pruessner and O'Leary [7]. This corresponds to a Tikhonov regularization of our structured total least squares problem and results in a fast solution algorithm for the problem considered in [5, 6, 7].

The core of the algorithm in [4], based on a more general algorithm of [8], relies on two results: the representation of the generators for the matrix M^TM that appears in the normal equations when A is Toeplitz, and then a fast factorization of a matrix derived from these generators. So we begin in Section 2 with a review of the problem formulation from [4, 8], and in Section 3, we derive the generators for M^TM when A

*This work was partially supported by the National Science Foundation under Grant CCR-0204084.

† Mathematics Department, University of Maryland, College Park, MD 20742 (kalsi@cs.umd.edu).

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

is any matrix of small displacement rank. In Section 4 we show that it is inexpensive to form a Cholesky factorization of $M^T M$ whenever Z_1 and Z_2 are lower triangular matrices. Section 6 concerns the generalization of this algorithm when a regularization constraint is to be applied. We show that in one formulation of such problems, the displacement rank of $M^T M$ is lower than expected. In Section 6, we apply this result to an important special case, image deblurring, and in Section 7 we present some numerical results.

2. Problem Formulation. Suppose that the matrix E can be specified by p parameters $\alpha_1, \ldots, \alpha_p$. For example, if E is a Toeplitz matrix, then

$$E = \begin{bmatrix} \alpha_n & \alpha_{n-1} & \ldots & \alpha_1 \\ \alpha_{n+1} & \alpha_n & \ldots & \alpha_2 \\ \ldots & \ldots & \ldots & \ldots \\ \alpha_{m+n-1} & \alpha_{m+n-2} & \ldots & \alpha_m \end{bmatrix},$$

and $p = m + n + 1$. We rewrite our problem as

$$\min_{\alpha,\beta,x} \left\| \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \right\|_2^2 \tag{2.1}$$

where

$$\beta = (A + E)x - y\,.$$

Following [4], we have replaced the term $\|E\|_F^2$ by $\alpha^T \alpha$, equivalent except for scaling of the entries α_i^2.

We define the matrix $X \in \mathcal{R}^{m \times p}$ by the equation

$$X\alpha = Ex\,.$$

For example, if E is Toeplitz, then $p = m + n - 1$ and

$$X = \begin{bmatrix} x_n & x_{n-1} & \ldots & x_1 & 0 & \ldots & \ldots & 0 \\ 0 & x_n & x_{n-1} & \ldots & x_1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & x_n & x_{n-1} & \ldots & \ldots & x_1 \end{bmatrix}$$

Following [8], we form a quadratic approximation to (2.1) by using linear approximations $\alpha + \Delta\alpha$ and $x + \Delta x$, resulting in

$$\begin{aligned} \beta &\approx (A + (E + \Delta E))(x + \Delta x) - y \\ &\approx (A + E)x + X\Delta\alpha + (A + E)\Delta x - y \end{aligned}$$

so that

$$\left\| \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} X & A + E \\ I & 0 \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta x \end{bmatrix} + \begin{bmatrix} (A + E)x - y \\ \alpha \end{bmatrix} \right\|_2^2$$

If we minimize this with respect to $\Delta\alpha$ and Δx, then we can form a new approximation

$$\begin{aligned} \alpha &= \alpha + \Delta\alpha \\ x &= x + \Delta x \end{aligned}$$

to the solution of (2.1) and then repeat the procedure until convergence. As noted by [8], this is a Gauss-Newton algorithm applied to (2.1) and although it is not guaranteed to converge to the global solution, it will at least find a local one.

Therefore, the main computational task is to solve linear least squares problems of the form

$$\min_{\Delta\alpha,\Delta x} \left\| M \begin{bmatrix} \Delta\alpha \\ \Delta x \end{bmatrix} + \begin{bmatrix} (A+E)x - y \\ \alpha \end{bmatrix} \right\|_2^2 \tag{2.2}$$

where

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

One way is to accomplish this is to solve the normal equations, the optimality conditions for this problem, and that involves solving the linear system

$$M^T M \begin{bmatrix} \Delta\alpha \\ \Delta x \end{bmatrix} = -M^T \begin{bmatrix} (A+E)x - y \\ \alpha \end{bmatrix} \tag{2.3}$$

We now derive the tools necessary to do this efficiently.

3. Generators for M^TM. Our first tool is the derivation of a generator for the matrix M^TM when M has low displacement rank.

3.1. The Displacement Rank of M^TM. Suppose that M has low displacement rank relative to the matrices $Z_1 \in \mathcal{R}^{(m+p)\times(m+p)}$ and $Z_2 \in \mathcal{R}^{(n+p)\times(n+p)}$, which means that if we define

$$N \equiv M - Z_1 M Z_2^T ,$$

then $\text{rank}(N) = \rho_1$, which is small relative to $n+p$.

Suppose

$$\tilde{Z} = Z_1 + W$$

is an orthogonal matrix ($\tilde{Z}^T\tilde{Z} = I$), where W has rank ρ_2, also assumed to be small. For example, if E is Toeplitz, let Z_1 be the shift-down matrix with ones on its subdiagonal and zeros elsewhere, and then W is the matrix with a one in the last position of row 1.

Then M^TM also has low displacement rank relative to Z_2, as we can see from the identity

$$\begin{aligned} M^TM - Z_2M^TMZ_2^T &= M^TM - Z_2M^T\tilde{Z}^T\tilde{Z}MZ_2^T \\ &= M^TM - (M - N - WMZ_2^T)^T(M - N - WMZ_2^T) \\ &= (N + WMZ_2^T)^T(M - N - WMZ_2^T) + M^T(N + WMZ_2^T) \end{aligned}$$

THEOREM 3.1. *If the rank of $N \equiv M - Z_1MZ_2^T$ is ρ_1 and if the orthogonal matrix $\tilde{Z}$ is equal to $Z_1 + W$ where W has rank ρ_2, then*

$$\begin{aligned} M^TM - Z_2M^TMZ_2^T = &-N^TN + N^T(M - WMZ_2^T) + (M^T - (WMZ_2^T)^T)N \\ &-(WMZ_2^T)^T(WMZ_2^T) + M^T(WMZ_2^T) + (WMZ_2^T)^TM \end{aligned}$$

has rank at most $2(\rho_1 + \rho_2)$.

Proof. The equation in the statement of the theorem is a regrouping of the terms in the previous equation. The rank of $N + WMZ_2^T$ is at most the rank of N plus the rank of W, so the rank of the sum in that equation is at most $2(\rho_1 + \rho_2)$. ☐

3.2. Deriving the Generators for the Toeplitz Example. For our Toeplitz example, we have

$$W = e_1 e_{m+p}^T \, .$$

Since $M - Z_1 M Z_2^T$ is nonzero only in rows 1 and $m+1$ and in columns 1 and $p+1$, then

$$N = M - Z_1 M Z_2^T = e_1 r_1^T - e_{m+1} r_m^T + e_{m+1} e_1^T + c_p e_{p+1}^T$$

where

$$\begin{aligned} r_1^T &= e_1^T M, \\ r_m^T &= e_{m+1}^T Z_1 M Z_2^T \, , \\ c_p &= M e_{p+1} - m_{1,p+1} e_1 \, . \end{aligned}$$

Note that $e_1^T c_p = e_{m+1}^T c_p = 0$.

We compute

$$W M Z_2^T = e_1 e_{m+p}^T M Z_2^T = e_1 e_{p+1}^T \, ,$$

and, since

$$\begin{aligned} e_1^T M &= r_1^T \, , \\ e_{m+1}^T M &= e_1^T \, , \end{aligned}$$

it is then clear from Theorem 3.1 that $M^T M - Z_2 M^T M Z_2^T$ is the sum of outer products of various vectors with only 5 different vectors: $r_1^T, r_m^T, e_1^T, e_{p+1}^T$, and $c_p^T M$, so the rank is 5.

It is useful to write the displacement in symmetric form. To do this for the Toeplitz example, we compute each of the terms in the Theorem:

$$\begin{aligned} -N^T N &= -r_1 r_1^T - r_m r_m^T - e_1 e_1^T \\ &\quad -(c_p^T c_p) e_{p+1} e_{p+1}^T + r_m e_1^T + e_1 r_m^T \\ N^T(M - W M Z_2^T) & \\ +(M^T - (W M Z_2^T)^T) N &= 2 r_1 r_1^T - r_m e_1^T + 2 e_1 e_1^T + e_{p+1} c_p^T M \\ &\quad - r_1 e_{p+1}^T - e_1 r_m^T + M^T c_p e_{p+1}^T - e_{p+1} r_1^T \\ -(W M Z_2^T)^T (W M Z_2^T) &= e_{p+1} e_{p+1}^T \\ M^T (W M Z_2^T) + (W M Z_2^T)^T M &= r_1 e_{p+1}^T + e_{p+1} r_1^T \end{aligned}$$

Adding these terms together, we obtain

$$\begin{aligned} M^T M - Z_2 M^T M Z_2^T &= e_1 e_1^T + r_1 r_1^T - r_m r_m^T (1 - c_p^T c_p) e_{p+1} e_{p+1}^T \\ &\quad + e_{p+1} c_p^T M + M^T c_p e_{p+1}^T \\ &= e_1 e_1^T + r_1 r_1^T - r_m r_m^T - M^T c_p (M^T c_p)^T / \gamma^2 \\ &\quad + (\gamma e_{p+1} + M^T c_p / \gamma)(\gamma e_{p+1} + M^T c_p / \gamma)^T \end{aligned}$$

where $\gamma^2 = (1 - c_p^T c_p)$.

4. Determining a Cholesky Factorization from the Generators. We now know how to determine ρ vectors g_i so that

$$M^T M - Z_2 M^T M Z_2^T = \sum_{i=1}^{\rho} s_i g_i g_i^T$$

where s_i equals plus or minus 1. When Z_1 and Z_2 are shift-down matrices, it has been shown [4, 1, 3] that this implies that

$$\begin{aligned} M^T M &= \sum_{i=1}^{\rho} s_i L_i L_i^T \\ &= \left[\begin{array}{ccc} L_1 & \ldots & L_\rho \end{array} \right] S \left[\begin{array}{c} L_1^T \\ \vdots \\ L_\rho^T \end{array} \right] \end{aligned}$$

where $S = \operatorname{diag}(s_i)$ and L_i is the lower triangular Toeplitz matrix with first row equal to g_i^T. We now generalize this result somewhat.

THEOREM 4.1. *If Z_1 is nilpotent, then*

$$A - Z_1 A Z_2^T = g h^T$$

if and only if

$$A = L_1(g) L_2^T(h)$$

where

$$L_i(x) = \left[\begin{array}{cccc} x & Z_i x & \ldots & Z_i^{n+p-1} x \end{array} \right] .$$

Proof. Suppose $A = L_1(g) L_2^T(h)$. Observe that

$$\begin{aligned} L_1(g) L_2^T(h) &= \left[\begin{array}{cccc} g & Z_1 g & \ldots & Z_1^{n+p-1} g \end{array} \right] \left[\begin{array}{c} h^T \\ h^T Z_2^T \\ \vdots \\ h^T (Z_2^T)^{n+p-1} \end{array} \right] \\ &= \sum_{j=0}^{n+p-1} Z_1^j g h^T Z_2^j \end{aligned}$$

and

$$Z_1 L_1(g) L_2^T(h) Z_2^T = \sum_{j=0}^{n+p-1} Z_1^{j+1} g h^T Z_2^{j+1}$$

so, since $Z_1^{n+p} = 0$, we conclude that

$$L_1(g) L_2^T(h) - Z_1 L_1(g) L_2^T(h) Z_2^T = g h^T .$$

To prove the converse, suppose $A - Z_1 A Z_2^T = g h^T$. Then, since

$$g h^T = L_1(g) L_2^T(h) - Z_1 L_1(g) L_2^T(h) Z_2^T ,$$

we conclude that if $E = A - L_1(g)L_2^T(h)$, then

$$E = Z_1 E Z_2^T \,.$$

Now since Z_1 is nilpotent, $Z_1^p = 0$ for some $p \le n$. Therefore, $Z_1^{p-1}E = Z_1^p E Z_2^T = 0$, and working backward in powers of Z_1, we see that $Z_1^0 E = Z_1 E Z_2^T = 0$, so $A = L_1(g)L_2^T(h)$. $\square$

The following corollary can be proved by finite induction.

Corollary 4.2. *If Z_1 is nilpotent, then*

$$A - Z_1 A Z_2^T = \sum_{i=1}^{\rho} g_i h_i^T$$

if and only if

$$A = \sum_{i=1}^{\rho} L_1(g_i) L_2^T(h_i)\,.$$

In order to solve our least squares problem, we wish to determine a Cholesky factorization

$$M^T M = L L^T\,,$$

so we need to reduce the matrix

$$\begin{bmatrix} L_1^T \\ \vdots \\ L_\rho^T \end{bmatrix}$$

to upper triangular form.

If Z_1 and Z_2 are shift-down matrices, then [4] shows how to do this reduction fast. Using our corollary, we see that this can be done fast whenever Z_1 and Z_2 are lower triangular matrices. We present the algorithm for this slightly more general case.

The algorithm proceeds by columns, putting zeros below the main diagonal. Note that

$$\hat{L} \equiv \begin{bmatrix} L_1^T \\ \vdots \\ L_\rho^T \end{bmatrix} = \begin{bmatrix} h_1^T \\ h_1^T Z_2^T \\ \dots \\ h_1^T (Z_2^T)^{n+p} \\ \vdots \\ h_\rho^T \\ h_\rho^T Z_2^T \\ \dots \\ h_\rho^T (Z_2^T)^{n+p} \end{bmatrix}$$

Suppose we determine a rotation between the first row h_1^T and row $n+p+1$, which contains h_2^T, to zero the first element of h_2^T. The same rotation between $h_1^T(Z_2^T)^j$ and $h_2^T(Z_2^T)^j$ $(j = 1, \dots, m+p-1)$ also zeroes the first element of $h_2^T(Z_2^T)^j$ since Z_2^T is upper triangular. Therefore, by introducing one zero into our matrix, we have implicitly introduced $m+p-1$ more, so we can put zeroes below the main diagonal in column 1 by using only $\rho-1$ rotations, independent of the size of $m+p$.

We then use the resulting second row, equal to the first row postmultiplied by Z_2^T, to zero the second element of row $n+p+1$. Again this implicitly introduces additional zeros, $m+p-2$ of them, and we complete the operations on column 2 by using $\rho-1$ rotations.

If we repeat this for each column, we accomplish our reduction.

Let G be the matrix whose rows are g_i^T. We can thus reduce $\hat{L}$ to upper triangular form just by operating on the matrix G.

We design our algorithm to use Givens rotations as often as possible, minimizing the number of hyperbolic rotations in order to preserve stability. We set

$$s_i = \begin{cases} 1 & \text{if } g_i = h_i, \\ -1 & \text{if } g_i = -h_i, \end{cases}$$

so that

$$A = \sum_{i=1}^{\rho} s_i L_1(g_i) L_2^T(g_i)\,.$$

A Givens rotation can be used between row i and row j whenever s_i and s_j have the same sign; if they have different signs, then we must use a hyperbolic rotation. We'll assume that we have ordered the generators so that the first $\hat{\rho}$ rows of G have $s_i = 1$ and the remaining ones have $s_i = -1$.

Algorithm Reduce(G)[1]

For $j = 1, \ldots, n+p$,
- For $i = 2, \ldots, \hat{\rho}$,
  - If g_{ij} is nonzero, then
    - zero it by a Givens rotation between row 1 and row i;
- end for
- For $i = \hat{\rho}+2, \ldots, \rho$,
  - If g_{ij} is nonzero, then
    - zero it by a Givens rotation between row $\hat{\rho}+1$ and row i;
- end for
- If $g_{\hat{\rho}+1,j}$ is nonzero, then
  - zero it by a stabilized hyperbolic rotation between row 1 and row $\hat{\rho}+1$;
- Then the jth row of L^T is g_1^T, the first row of the current G matrix.
- Replace the first row of G by $g_1^T Z_2^T$ to form the pivot row for the next value of j.

end for

The cost of this reduction is at most $O(\rho(n+p)^2)$, ignoring sparsity, plus the cost of the multiplications by Z_2. Without exploiting the structure of $\hat{L}$ the cost would be

[1]There is an analogous algorithm, `FTriang`, in [4], for the special case in which A is Toeplitz, but it has some typographical errors. In the statement following "if i<m", g3 on the left-hand side should be g4. In 12 places on p. 552, "m+n" should be "mn1". Also, the numbering of the phases of the computation is off by one compared with the description in the paper ("Initialization" should be "Phase 1", etc.)

$O(\rho(n+p)^3)$. Once the factors LL^T are computed, they can then be used to solve (2.3).

5. Regularized Solutions. In many deblurring problems and other discretized problems involving integral equations of the first kind, the matrix A is so ill-conditioned that noise in the observations y is magnified in solving the STLS problem and a meaningful solution cannot be obtained.

In this case it is necessary to add a *regularization constraint* to the problem. One common regularization constraint is to restrict the size of the solution, or some linear transformation of the solution:

$$\|Cx\| \leq u$$

where u is a given scalar and C is commonly chosen to be the identity matrix or a difference operator. If C has low displacement rank relative to Z_1 and Z_2, then our algorithm can be easily modified to incorporate regularization. In this case, our problem (2.1) can be reformulated as

$$\min_{\alpha,\beta,x} \left\| \begin{bmatrix} \beta \\ \alpha \\ \lambda Cx \end{bmatrix} \right\|_2^2 \tag{5.1}$$

where $\beta = (A+E)x - y$ and λ, the regularization parameter, is the Lagrange multiplier for the new constraint. Using a derivation similar to that above, the linearization of (5.1) results in the following problem to be solved at each step of the iteration:

$$\min_{\Delta\alpha,\Delta x} \left\| \begin{bmatrix} X & A+E \\ I & 0 \\ 0 & \lambda C \end{bmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} + \begin{pmatrix} \beta \\ \alpha \\ \lambda Cx \end{pmatrix} \right\|_p .$$

Thus, our new M matrix is the old matrix M augmented by the extra rows $[0, \lambda C]$, and the only change necessary in the algorithm is to find the generators of this matrix rather than the old one.

The displacement structure of this matrix is greatly simplified if C is upper triangular and Z_2 is the shift-down matrix. As noted before, W is zero except for a one in the last position of the first row, and thus WM is zero except for a λ in the last position of the first row. Therefore, $WMZ_2^T = 0$, so, applying Theorem 3.1, we have the following result.

THEOREM 5.1. *If C is upper triangular and Z_2 is the shift-down matrix, then*

$$M^TM - Z_2M^TMZ_2^T = (M - N/2)N^T + N(M - N/2)^T$$

and has rank $2\rho_1$, where ρ_1 is the rank of N.

Using the identity

$$ab^T + ba^T = \frac{1}{2}(a+b)(a+b)^T - \frac{1}{2}(a-b)(a-b)^T,$$

we can easily symmetrize the generators.

6. Application to Image Deblurring. Consider the problem of deblurring images whose point-spread function is spatially invariant. In this case, we have measured a set of values

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mm} \end{bmatrix}$$

and want to reconstruct an image

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix}$$

when the matrix A is block Toeplitz with Toeplitz blocks.

Let us order the pixels by rows to create a one-dimensional vector of unknowns:

$$x = [x_{11}, x_{12}, \ldots, x_{1n}, \ldots, x_{n1}, x_{n2}, \ldots, x_{nn}]^T ,$$

and similarly, we create a vector y of observations.

For definiteness, we'll assume that the blurring function averages the $p^2 = 9$ nearest neighbors of each pixel, and that $m = n + p - 1$. In this case, the matrix A has three block diagonals, each with three diagonals:

$$A = \begin{bmatrix} T_1 & & & & \\ T_2 & T_1 & & & \\ T_3 & T_2 & T_1 & & \\ & \ddots & \ddots & \ddots & \\ & & T_3 & T_2 & T_1 \\ & & & T_3 & T_2 \\ & & & & T_3 \end{bmatrix},$$

$$T_j = \begin{bmatrix} t_{j1} & & & & \\ t_{j2} & t_{j1} & & & \\ t_{j3} & t_{j2} & t_{j1} & & \\ & \ddots & \ddots & \ddots & \\ & & t_{j3} & t_{j2} & t_{j1} \\ & & & t_{j3} & t_{j2} \\ & & & & t_{j3} \end{bmatrix}, \quad j = 1, 2, 3.$$

The dimension of A is $m^2 \times n^2$, and the dimension of T_j is $m \times n$.

The matrix E has the same structure as A, but with entries α_{ji}, and the relation $X\alpha = Ex$ holds if we define

$$X = \begin{bmatrix} X_1 & 0 & 0 \\ X_2 & X_1 & 0 \\ X_3 & X_2 & X_1 \\ \vdots & \vdots & \vdots \\ X_n & X_{n-1} & X_{n-2} \\ 0 & X_n & X_{n-1} \\ 0 & 0 & X_n \end{bmatrix}$$

with

$$X_j = \begin{bmatrix} x_{j1} & 0 & 0 \\ x_{j2} & x_{j1} & 0 \\ x_{j3} & x_{j2} & x_{j1} \\ \vdots & \vdots & \vdots \\ x_{jn} & x_{j,n-1} & x_{j,n-2} \\ 0 & x_{jn} & x_{j,n-1} \\ 0 & 0 & x_{jn} \end{bmatrix}$$

The matrix X has dimension $m^2 \times p^2$, with X_j of dimension $m \times p$.

The displacement rank of the resulting

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

(with $C = I$) is $2m$, since the matrix $M - Z_1 M Z_2$ has nonzeros in rows $1, m+1, \ldots, m^2+1$, and in columns $1, p+1, \ldots, p^2+1$ and then every nth column after that. Using Theorem 5.1, we see that $M^T M$ has displacement rank $4m$.

The bulk of the work in the algorithm is in factoring $M^T M$ using its generators. Factorization requires $O((n^2+p^2)^2)$ rotations, with $O(n^2+p^2)$ multiplications each. Thus the work is proportional to the number of pixels raised to the 2.5 power. It is possible to save work by refactoring $M^T M$ less frequently and using an iterative method, preconditioned by the most recent factorization, to compute the direction. Fu and Barlow have also developed an iterative method for solving this system [2].

7. Numerical Results. We demonstrate the RSTLS algorithm on a small image deblurring problem.

Figure 7.1 shows the original and blurred image. The point-spread function was Gaussian with $p = 5$. The noise added to each element of the blurred image and the point spread function was normally distributed with mean zero and standard deviation $\sigma = 0.05$.

We compare 5 algorithms:

- RSTLS, with $\lambda = 3\sigma$. The iteration was terminated when the norm of the change in the image was less than .001.
- Tikhonov regularization, with the same value of λ.
- Truncated SVD, dropping singular values smaller than 3σ.
- Truncation based on the ℓ_∞ norm, ensuring that components of the answer are no greater than greater than 2 in absolute value.
- TLS approximated by the Lanczos algorithm. The iteration was terminated when the norm of the computed image grew larger than the norm of the true image.

The results of the algorithms are shown in Figures 7.1 and 7.2. Lanczos took 41 iterations, while RSTLS took 78 costlier iterations.

Figures 7.3 and 7.4 show the results for a lower noise level: $\sigma = 0.01$. Lanczos took 78 iterations, while RSTLS took 10.

The 2-norm of the relative error in the computed images for both noise levels are tabulated in Table 7. The relative errors agree with the “eye-norm” errors measured by human judgement of the images: RSTLS produces the best result, with the TLS algorithm producing the second best.

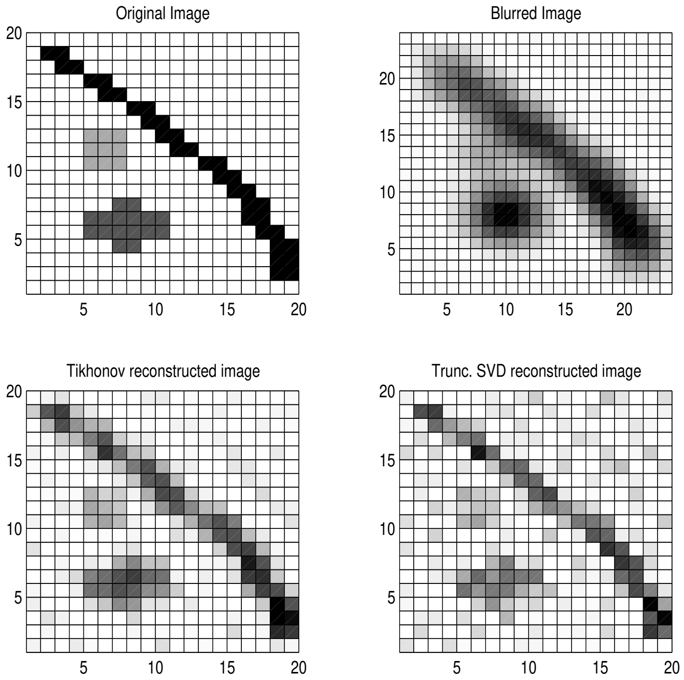

FIG. 7.1. *Original image, blurred image, and results of algorithms when noise level is* $\sigma = 0.05$.

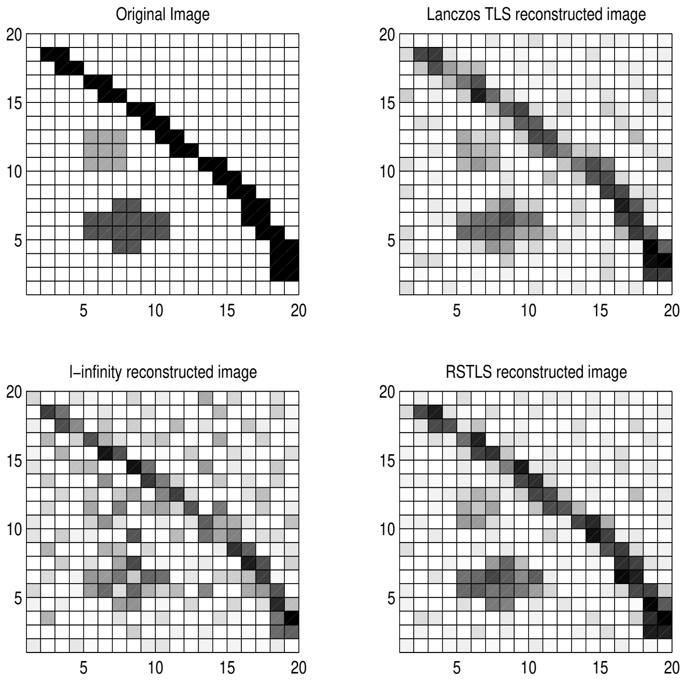

FIG. 7.2. *Original image, image, and results of algorithms when noise level is* $\sigma = 0.05$.

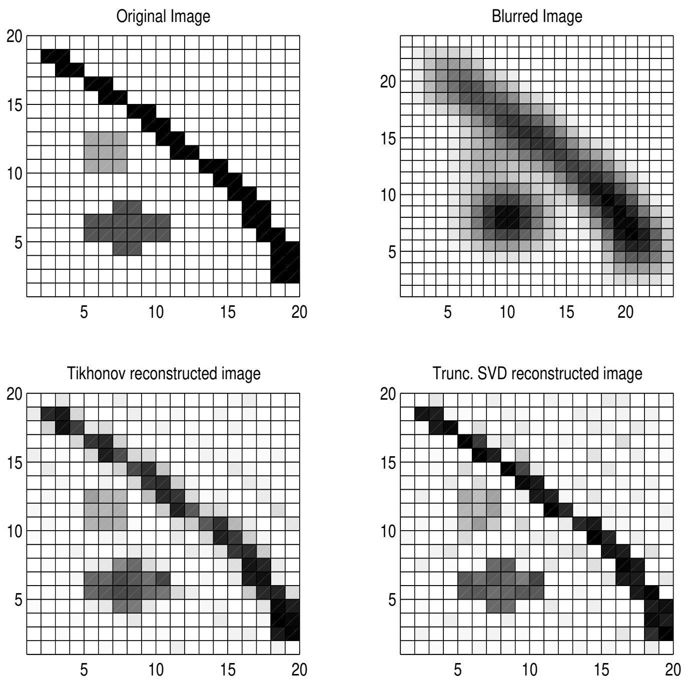

FIG. 7.3. *Original image, blurred image, and results of algorithms when noise level is* $\sigma = 0.01$.

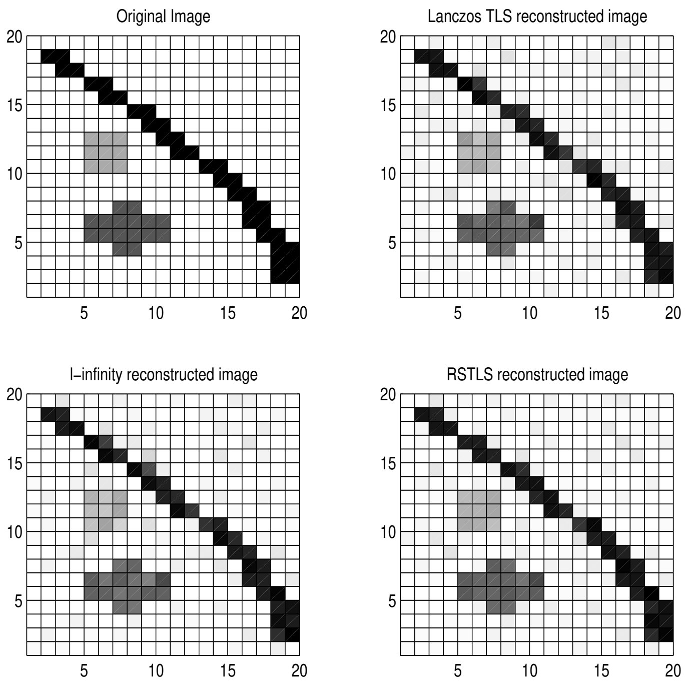

FIG. 7.4. *Original image, blurred image, and Results of algorithms when noise level is* $\sigma = 0.01$.

Algorithm	$\sigma = .05$	$\sigma = .01$
ℓ_∞	0.675	0.131
TSVD	0.417	0.152
Tikhonov	0.395	0.237
Lanczos TLS	0.369	0.133
RSTLS	0.292	0.113

TABLE 7.1
Relative error in the reconstructed images.

8. Conclusions. We have derive the generators for $M^T M$ when M is any matrix of small displacement rank. We have shown that it is inexpensive to form a Cholesky factorization of $M^T M$ whenever Z_1 and Z_2 are lower triangular matrices, and we have generalized this algorithm when a regularization constraint is to be applied. We have shown that the algorithm can be applied to deblurring of small images. Future work will focus on using the displacement rank results to speed up the ℓ_∞ and ℓ_1 algorithms of [7].

REFERENCES

[1] J. CHUN, T. KAILATH, AND H. LEV-ARI, *Fast parallel algorithms for QR and triangular factorizations*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 899–913.

[2] H. FU AND J. BARLOW, *A regularized structured total least squares algorithm for high resolution image reconstruction*, tech. report, Computer Science and Engineering Department, Pennsylvania State University, August 2002.

[3] T. KAILATH, S. KUNG, AND M. MORF, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl., 68 (1979), pp. 395–407.

[4] N. MASTRONARDI, P. LEMMERLING, AND S. V. HUFFEL, *Fast structured total least squares algorithm for solving the basic deconvolution algorithm*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 533–553.

[5] V. MESAROVIĆ, N. GALATSANOS, AND A. KATSAGGELOS, *Regularized constrained total least squares image restoration*, IEEE Transactions on Image Processing, 4 (1995), pp. 1096–1108.

[6] M. K. NG, R. J. PLEMMONS, AND F. PIMENTEL, *A new approach to constrained total least squares image restoration*, Linear Algebra and Its Appl., 316 (2000), pp. 237–258.

[7] A. PRUESSNER AND D. P. O'LEARY, *Blind deconvolution using a regularized structured total least norm algorithm*, SIAM J. on Matrix Anal. Appl., to appear.

[8] 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.