TOTAL LEAST NORM FORMULATION AND SOLUTION FOR STRUCTURED PROBLEMS

J. BEN ROSEN*, HAESUN PARK† AND JOHN GLICK‡

Abstract. A new formulation and algorithm is described for computing the solution to an overdetermined linear system, $Ax \approx b$, with possible errors in both A and b. This approach preserves any affine structure of A or $[A \mid b]$, such as Toeplitz or sparse structure, and minimizes a measure of error in the discrete L_p norm, where $p = 1, 2$ or ∞. It can be considered as a generalization of Total Least Squares and we call it Structured Total Least Norm(STLN).

The STLN problem is formulated, the algorithm for its solution is presented and analyzed, and computational results are summarized which illustrate the algorithm convergence and performance on a variety of structured problems. For each test problem, the solutions obtained by Least Squares, Total Least Squares, and STLN with $p = 1, 2$, and ∞ were compared. These results confirm that the STLN algorithm is an effective method for solving problems where A or b has a special structure, or where errors can occur only in some of the elements of A and b.

Key words. data fitting, Hankel structure, least squares, linear prediction, minimization, overdetermined linear systems, Toeplitz structure, structured total least norm, total least squares, 1-norm, 2-norm, ∞-norm

1. Formulation of Structured Total Least Norm (STLN) Problems.

An important data fitting technique developed over the past 15 years is that of Total Least Squares (TLS) [7, 8]. The TLS method is a generalization of the least squares method for an overdetermined system of linear equations, $Ax \approx b$, where A is $m \times n$, with $m > n$. In the least squares solution it is assumed that the matrix A is known without error, but that the vector b is subject to error. The vector x is determined so that $\|b - Ax\|_2 = \min$.

TLS allows the possibility of error in the elements of a given (data) matrix A, so that the modified matrix is given by $A + E$, where E is an error matrix to be determined. The TLS problem can then be stated as that of finding E and x, such that

$$\|E \mid r\|_F = \min, \tag{1.1}$$

where $r = b - (A + E)x$, and $\| \cdot \|_F$ represents the Frobenius matrix norm.

A complete description of TLS is given in the recent book [14], where many applications to signal processing, system identification and system response prediction are described. In many of these applications the matrix A has a special structure, such as Toeplitz structure, or is a large, sparse matrix with relatively few nonzero elements.

* Computer Science Department, University of Minnesota, Minneapolis, MN 55455, and Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093 (e-mail: rosen@cs.umn.edu, jbrosen@cs.ucsd.edu). The work of the first and the third authors was supported in part by the Air Force Office of Scientific Research grant AFOSR-91-0147 and the Minnesota Supercomputer Institute.

† Computer Science Department, University of Minnesota, Minneapolis, MN 55455, U.S.A. (e-mail: hpark@cs.umn.edu). The work of this author was supported in part by the National Science Foundation grant CCR-9209726 and also by a contract DAAL02-89-C-0038 between the Army Research Office and the University of Minnesota for the Army High Performance Computing Research Center.

‡ Department of Mathematics and Computer Science, University of San Diego, San Diego, CA 92110 (e-mail: glick@teetot.acusd.edu).

Furthermore, in some applications, errors occur only in a small number of the elements of A, so that while A may be dense, the matrix E could be sparse.

The generally used computational method for solving TLS is based on the singular value decomposition (SVD) of $[A\,|\,b]$. A complete discussion of efficient computational methods for solving TLS based on SVD is given in Chapter 4 of [14]. For applications where the matrix A has a special structure, the SVD based methods may not always be appropriate, since they do not preserve the special structure. In fact, using the SVD approach the matrix E will typically be dense, with no special structure, even when A is Toeplitz or sparse. Thus, even those elements of E which should remain zero, will typically become non-zero. Also in some situations, the use of a norm other than Frobenius norm may be preferable. For example, if the data contains outliers, an L_1 norm might be more suitable.

A new approach, to be described, is called Structured Total Least Norm (STLN), and it addresses these situations. The STLN formulation allows other norms, in addition to the Frobenius norm, to be used. In particular, the problem can be formulated so as to minimize the error in either the L_1 norm or the L_∞ norm, in addition to the Frobenius norm used in TLS. Another important advantage of the STLN formulation is that it permits a known structure of the matrix A and $[A\,|\,b]$ to be preserved in $A+E$ and $[A+E\mid b+r]$, respectively. Requirements of this kind occur in important applications. For example, a Toeplitz structure occurs in system identification problems [4, 6] and in frequency estimation [1, 11]. For other applications, see [1, 4, 14, 17].

The new approach will guarantee that, for example, E will have the same Toeplitz structure as A, or that only those elements of E which represent possible errors in A, are permitted to be non-zero. In general, it can preserve any given affine structure in the computed error matrix E. Restrictions of this type have been imposed by the use of additional constraints on the problem, in some earlier papers [3, 13]. The use of the L_1 norm for this kind of problem has also been investigated [10], using a different method than the one presented here. This extension of the TLS solution to incorporate the algebraic pattern of the errors in A is also studied in [1] and [4] as "Constrained TLS" and "Structured TLS", respectively, both for the L_2 norm only. In [1], a complex Newton's method is utilized to solve the problem, whereas in [4], nonlinear singular value decomposition is defined and an algorithm to compute the solution is derived. For the comparison of these two algorithms, see [15]. The approach in our algorithm is different from these two and we will present the comparison of these three algorithms elsewhere.

Our formulation for solving the STLN problem takes full advantage of the special structure of a given matrix A. In particular, when $q(\leq mn)$ elements of $A \in \Re^{m\times n}$ are subject to error, a vector $\alpha \in \Re^{q\times 1}$ is used to represent the corresponding elements of the error matrix E. Note that for a sparse matrix, $q \ll mn$. Furthermore, if many elements of E must have the same value, then q is the number of *different* such elements. For example, in a Toeplitz matrix, each diagonal consists of elements with the same value, so $q \leq m+n-1$.

The vector α and the matrix E are equivalent, in the sense that given E, α is known, and vice versa. The matrix E is specified by those elements of A which may be subject to error. Each different non-zero element of E corresponds to one of the α_k, $k=1,\ldots,q$. Also the residual vector $r = b-(A+E)x$, is now a function of α and x, so $r = r(\alpha, x)$. Let D be a $(q\times q)$ diagonal weighting matrix that accounts for the repetition of elements of α in the matrix E. Then the STLN problem can be stated as follows:

$$\min_{\alpha,x} \left\| \begin{array}{c} r(\alpha,x) \\ D\alpha \end{array} \right\|_p, \tag{1.2}$$

where $\|\cdot\|_p$ is the vector p-norm, for $p = 1, 2,$ or ∞.

For $p = 2$, and a suitable choice for D, the problem (1.2) is equivalent to the TLS problem (1.1), with the additional requirement that the structure of A must be preserved by $A + E$.

2. STLN Algorithm. An iterative algorithm for solving the STLN problem will now be described. To do this, we first explain the relationship between $E \in \Re^{m\times n}$ and $\alpha \in \Re^{q\times 1}$. Specifically, the vector Ex must be represented in terms of α. This is accomplished by defining a matrix $X \in \Re^{m\times q}$ such that

$$X\alpha = Ex. \tag{2.1}$$

The elements of X consist of the elements of $x \in \Re^{n\times 1}$, with suitable repetition, giving X a special structure. The number of nonzero elements in both E and X will be equal, so that if E is sparse, X will also be sparse. Furthermore, if the non-zero elements α_k, $k = 1, \ldots, q$, of E are properly ordered, then X will have a similar structure to E: for example, if E is a Toeplitz matrix, then X will also be a Toeplitz matrix. The construction of X from E is described in Section 3.

The minimization required by (1.2) is done by using a linear approximation to $r(\alpha, x)$. Let Δx represent a small change in x, and ΔE represent a small change in the variable elements of E. From (2.1), we have

$$X\Delta\alpha = (\Delta E)x, \tag{2.2}$$

where $\Delta\alpha$ represents the corresponding small change in the elements of α. Then, neglecting the second order terms in $\|\Delta\alpha\|$ and $\|\Delta x\|$,

$$\begin{aligned} r(\alpha + \Delta\alpha, x + \Delta x) &= b - (A + E)x - X\Delta\alpha - (A + E)\Delta x \\ &= r(\alpha, x) - X\Delta\alpha - (A + E)\Delta x. \end{aligned} \tag{2.3}$$

The linearization of (1.2) now becomes:

$$\min_{\Delta\alpha,\Delta x} \left\| \begin{bmatrix} X & A+E \\ D & 0 \end{bmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} + \begin{pmatrix} -r \\ D\alpha \end{pmatrix} \right\|_p. \tag{2.4}$$

To start the iterative algorithm, the initial values of $E = 0$, and the least norm value of $x = x_{ln}$ are used, where x_{ln} is given by:

$$\min_x \|b - Ax\|_p. \tag{2.5}$$

Note that the initial x for $p = 2$ is the solution to the corresponding least squares problem.

The Structured Total Least Norm (STLN) algorithm is summarized in Algorithm STLN. The computational method by which Step 2a is carried out depends on the

---

Algorithm STLN

Input – A Structured Total Least Norm problem (1.2), with specified matrices A, D, vector b, and tolerance ϵ.

Output – Affine structured error matrix E, vector x, and STLN error.

Begin

1. Set $E = 0$, $\alpha = 0$, compute x from (2.5) and X from x, and set $r = b - Ax$.
2. **repeat**
   (a) $\underset{\Delta x, \Delta\alpha}{\text{minimize}} \left\| \begin{bmatrix} X & A+E \\ D & 0 \end{bmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} + \begin{pmatrix} -r \\ D\alpha \end{pmatrix} \right\|_p .$
   (b) Set $x := x + \Delta x$, $\alpha := \alpha + \Delta\alpha$.
   (c) Construct E from α, and X from x. Compute $r = b - (A+E)x$.
   until $(\|\Delta x\|, \|\Delta\alpha\| \le \epsilon)$

End

---

value of p. For $p = 2$, the corresponding least squares problem is solved efficiently by a QR factorization of the matrix

$$M = \begin{bmatrix} X & A+E \\ D & 0 \end{bmatrix} \tag{2.6}$$

when $A + E$ has full column rank, since M has full column rank in this case. In some applications, the right hand side vector is structured as well. For example, in linear prediction, either $[A\,|\,b]$ or $[b\,|\,A]$ follows the Toeplitz or Hankel structure. We also show how the STLN algorithm can be modified to handle this situation in Section 6. For $p = 1$, or $p = \infty$, Step 2a is solved as a linear program which takes advantage of the special structure of M (see Section 5). Since the matrix X has a special structure like $A + E$, the matrix M is also highly structured. To make Algorithm STLN efficient, it will be important to take advantage of the structure of M each time Step 2a is solved. For example, it is shown in [12] how a fast triangularization of M can be carried out when A is Toeplitz.

A theoretical justification for the STLN algorithm for $p = 2$, is presented in Section 4, and its computational performance illustrated in Section 7. In the next section the construction of the matrix X, given the structure of E, is described.

3. Construction of Matrix X. The matrix E is specified by those elements of A which may be subject to error. Each different non-zero element of E corresponds to one of the α_k, $k = 1, \ldots, q$, where the vector $\alpha = (\,\alpha_1 \cdots \alpha_q\,)^T$ represents $q(\le mn)$ elements of A which are subject to error. The order in which the α_k are numbered will affect the structure of the matrix X, but for any specified ordering, the structure of X is uniquely determined.

The construction of X (starting with a zero matrix) is carried out according to the following rule:

If α_k is the $(i,j)^{th}$ element of E, then x_j is the $(i,k)^{th}$ element of X, where $i = 1, \ldots, m$, $j = 1, \ldots, n$, and $k = 1, \ldots, q$.

For example, when

$$E = \begin{pmatrix} \alpha_2 & \alpha_1 & 0 \\ \alpha_3 & \alpha_2 & \alpha_1 \\ \alpha_4 & \alpha_3 & \alpha_2 \\ 0 & \alpha_4 & \alpha_3 \end{pmatrix}, \quad \text{we have} \quad X = \begin{pmatrix} x_2 & x_1 & 0 & 0 \\ x_3 & x_2 & x_1 & 0 \\ 0 & x_3 & x_2 & x_1 \\ 0 & 0 & x_3 & x_2 \end{pmatrix} \text{with } \alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix},$$

where only four diagonals of the Toeplitz matrix E are subject to error. For the sparse matrix

$$E = \begin{pmatrix} 0 & \alpha_1 & \alpha_2 \\ 0 & \alpha_3 & 0 \\ \alpha_4 & 0 & \alpha_3 \\ 0 & 0 & \alpha_5 \\ \alpha_4 & \alpha_2 & 0 \end{pmatrix}, \quad \text{we have} \quad X = \begin{pmatrix} x_2 & x_3 & 0 & 0 & 0 \\ 0 & 0 & x_2 & 0 & 0 \\ 0 & 0 & x_3 & x_1 & 0 \\ 0 & 0 & 0 & 0 & x_3 \\ 0 & x_2 & 0 & x_1 & 0 \end{pmatrix} \quad \text{with } \alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ a_5 \end{pmatrix},$$

where only nonzero elements of E are subject to error and the elements denoted with the same α_i are to be perturbed to have same values.

It is also useful to define $(q \times n)$ matrices P_i, $i = 1, \ldots, m$, as follows:

If α_k is the $(i,j)^{th}$ element of E, then the $(k,j)^{th}$ element of P_i is a one. All elements of P_i not equal to one, are zero.

Note that at most one element of any column of P_i is a one, and many columns of P_i may consist of all zeros. See Section 7 for some numerical examples.

With these definitions of the matrices X and P_i, it is easy to show that:

$$E = \begin{bmatrix} \alpha^T P_1 \\ \alpha^T P_2 \\ \vdots \\ \alpha^T P_m \end{bmatrix} \quad \text{and} \quad X = \begin{bmatrix} x^T P_1^T \\ x^T P_2^T \\ \vdots \\ x^T P_m^T \end{bmatrix}. \tag{3.1}$$

The relation (2.1) follows directly from (3.1).

4. STLN Optimality Conditions and Newton's Method. We now consider the two-norm case ($p = 2$) in more detail, since it has special properties which make a more complete theoretical analysis possible. For $p = 2$, the STLN problem (1.2) can be stated in terms of minimizing the differentiable function

$$\begin{aligned} \varphi(\alpha, x) &= \tfrac{1}{2}\|r(\alpha, x)\|_2^2 + \tfrac{1}{2}\|D\alpha\|_2^2 \\ &= \tfrac{1}{2}r^T r + \tfrac{1}{2}\alpha^T D^2 \alpha, \end{aligned} \tag{4.1}$$

where, $r = b - (A + E)x$.

The first-order optimality conditions for a local optimum of $\varphi(\alpha, x)$ are the vanishing of the gradients $\nabla_\alpha \varphi$ and $\nabla_x \varphi$. Using the relations presented in the previous section these conditions become:

$$\begin{aligned} \nabla_\alpha \varphi = -X^T r + D^2 \alpha = 0 \\ \nabla_x \varphi = -(A + E)^T r = 0. \end{aligned} \tag{4.2}$$

Now consider the least-squares solution of the $(m+q)$ equations in Step 2a of Algorithm STLN. The corresponding normal equations are:

$$M^T M \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} = M^T \begin{pmatrix} r \\ -D\alpha \end{pmatrix} = - \begin{pmatrix} \nabla_\alpha \varphi \\ \nabla_x \varphi \end{pmatrix}, \tag{4.3}$$

where the last equality follows directly from (4.2).

When the matrix M has full rank, the matrix $M^T M$ is positive definite, and (4.3) always has a unique solution for $(\Delta\alpha^T, \Delta x^T)$. This vector will be zero if, and only if, the right-hand-side of (4.3) vanishes. This means that convergence of Algorithm STLN (i.e., $(\Delta\alpha^T, \Delta x^T) \approx 0$) is equivalent to satisfying the optimality conditions (4.2).

We now show that Step 2a of Algorithm STLN is essentially Newton's method applied to the gradient of $\varphi(\alpha, x)$. To simplify notation, let $y^T = (\alpha^T, x^T)$, $\varphi(y) = \varphi(\alpha, x)$, and

$$\nabla\varphi(y) = \begin{pmatrix} \nabla_\alpha \varphi(\alpha, x) \\ \nabla_x \varphi(\alpha, x) \end{pmatrix}.$$

Let $H(y)$ be the Hessian of $\varphi(y)$. We wish to find y^*, such that $\nabla\varphi(y^*) = 0$.

An iteration of Newton's method to do this is given by:

$$\begin{aligned} &H(y)\Delta y = -\nabla\varphi(y) \\ &y := y + \Delta y. \end{aligned} \tag{4.4}$$

For $H(y)$ positive definite, and an initial y sufficiently close to $y*$, this Newton's method will converge to y^* at a second order rate. See, for example, Theorem 3.1.1 in [5].

To show the relationship between Step 2a of Algorithm STLN and (4.4) we note that the right-hand-sides of (4.3) and (4.4) are identical. Furthermore, it can be shown, using the relations in Section 3, that

$$H(y) = M^T M - \begin{pmatrix} 0 & P(r) \\ P^T(r) & 0 \end{pmatrix}, \tag{4.5}$$

where

$$P(r) = \sum_{i=1}^{m} r_i P_i,$$

is a matrix with norm $0(\|r\|)$.

Thus Step 2a is, in effect, a Gauss-Newton method which uses $M^T M$ as a positive definite approximation to $H(y)$ (see, for example, Section 6.1 in [5]). Computational experience with the STLN algorithm (Section 7) demonstrates that this is an effective strategy for this type of problem.

The differentiable function $\varphi(\alpha, x)$ we wish to minimize for $p = 2$, is not a convex function of (α, x). Therefore, there is no guarantee that a point satisfying the first-order optimality conditions (4.2) is a global minimum. In fact, it could be any stationary point of $\varphi(\alpha, x)$. In general, the Gauss-Newton method will converge to the closest local minimum when the residual $r(\alpha, x)$ is sufficiently small [5]. When there is no structure imposed on E, the STLN formulation (1.2) is equivalent to the TLS problem (1.1). In our preliminary test results, we have observed that the solution produced by the STLN was always the same as that from the TLS, when no structure is imposed on E. However, there is no theoretical guarantee that Algorithm STLN will produce the global minimum solution that the TLS via SVD produces. We do not propose Algorithm STLN for solving unstructured problems since the computational complexity will be high due to the large number of elements in α, which will be mn.

In many applications, the minimum residual is zero or very small when we have the exact data, and the global minimum value of α will be of the order of the noise in the matrix A. Therefore, the initial value $\alpha = 0$ is close to the global minimum value and convergence to the global minimum can often be expected. Our computational results confirm this expectation (see Section 7).

The function being minimized by STLN (given by (1.2)) for $p = 1, \infty$, is not differentiable, so the Gauss-Newton theory does not apply. Theoretical results on the convergence of the STLN algorithm for $p = 1, \infty$, are not known at this point, but are the subject of our continuing research.

5. STLN for $p = 1$ and $p = \infty$. For $p = 1$ or ∞, Step 2a is solved as a linear program (LP). To illustrate this, the linear program for $p = \infty$ is now summarized. The formulation for $p = 1$ is similar.

A scalar σ, representing the maximum norm is introduced, and the corresponding linear program is then given by:

$$\begin{array}{ll} \underset{\Delta\alpha,\Delta x,\sigma}{\text{minimize}} & \sigma \\ \text{subject to} & \begin{array}{l} -\sigma e_m \leq X\Delta\alpha + (A+E)\Delta x - r \leq \sigma e_m \\ -\sigma e_q \leq D\Delta\alpha + D\alpha \leq \sigma e_q \end{array} \end{array} \tag{5.1}$$

where $e_k \in \Re^{k\times 1}$ is the vector with every element equal to one. Note that a feasible solution to this problem is easily given ($\Delta\alpha = 0$, $\Delta x = 0$, σ sufficiently large), and since $\sigma \geq 0$, an optimal solution always exists. In this form the problem has more inequality constraints, $2(m + q)$, than variables, $n + q + 1$, so it is more efficient to consider (5.1) as the dual problem, and solve the equivalent primal.

The equivalent primal is:

$$\begin{array}{ll} \underset{y^{(i)}\geq 0}{\text{minimize}} & r^T y^{(1)} + \alpha^T D y^{(2)} - r^T y^{(3)} - \alpha^T D y^{(4)} \\ \text{subject to} & \begin{bmatrix} M^T & -M^T \\ e_{m+q}^T & e_{m+q}^T \end{bmatrix} \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ y^{(4)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \end{array} \tag{5.2}$$

The optimal solution and basis to (5.2) immediately gives the optimal dual vector $(\Delta\alpha, \Delta x, \sigma)$. Any available LP package (Simplex or Interior) can be used to solve (5.2), however it should be possible to use the special structure of M to solve (5.2) more efficiently. Another aspect of (5.2) which can be used to advantage is that only relatively small changes occur in the cost vector coefficients and the matrix M at each iteration of the STLN algorithm. Therefore the previous basis will often be optimal, or almost optimal, after the initial LP solution. An additional benefit obtained from the primal-dual relationship is information about the sensitivity of the STLN solution to changes in the data. Properly interpreted, the primal variables (elements of the vectors $y^{(i)}$) are measures of the change in the value of the minimum norm as a result of changes in the problem data.

It should also be noted that by the addition of one row and two columns to (5.2) a specified bound δ can be imposed, so that

$$\|D\alpha\|_\infty \leq \delta. \tag{5.3}$$

With this addition, the original STLN problem (1.2), with $p = \infty$, is modified so that α is limited to values satisfying (5.3). This restriction may be important in some applications: for example, if it is known that the errors in A cannot exceed some specified bounds. For the limiting case of $\delta = 0$, the solution to (1.2) and (5.3) will give the Least Norm solution (2.5) for $p = \infty$, while for sufficiently large δ it will give the STLN∞ solution to (1.2). For small values of δ, the solutions to (1.2) and (5.3) may differ significantly from the STLN∞ solution. Similar bounds can readily be imposed in the L_1 norm case.

6. STLN for Structured Vector b. In many applications, the structure is imposed not only on the data matrix A but also on the right hand side vector b or even on $[A \mid b]$. For example, in the least squares linear prediction problem, we need to solve

$$\min_x \|Ax - b\|_2 \tag{6.1}$$

where $A \in \Re^{m\times n}$ is a Toeplitz matrix with $m \geq n$ and the right hand side vector b follows the pattern of A, so that either $[A \mid b]$ is Toeplitz in backward prediction or $[b \mid A]$ is Toeplitz in forward prediction. For details on the least squares linear prediction problem, see [9]. In this section, we show how to modify Algorithm STLN so that it can treat possible errors in some (or all) elements of b in the same manner as errors in A are treated. We will discuss the Toeplitz structure in detail since it appears in numerous applications in signal processing, image processing, and system identification [1, 4, 9, 15]. The results presented in this section on Toeplitz structure apply to Hankel structure in a straightforward manner, since Hankel structure can be transformed to Toeplitz structure via permutations.

We introduce a vector β representing possible errors in selected elements of b. This is similar to α representing errors in A. Suppose different errors can occur in $q_2(\leq m)$ elements of b, specifically, in the elements b_{i_j}, $j = 1, \cdots, q_2$. The error vector $\beta \in \Re^{q_2\times 1}$ represents the error in b_{i_j}, $j = 1, \cdots, q_2$. The relation between β and b is given by a matrix $P_0 \in \Re^{m\times q_2}$, so that error in b is the same as $P_0\beta$. The matrix P_0 consists of only zeros and ones: the element P_{ij} of P_0 is one if β_j is the error in b_i, otherwise, it is zero. Note that every column of P_0 contains exactly one nonzero element.

Initially, E, α and β are all zero, and the new residual $\hat{r} = r = b - Ax$. In general,

$$\hat{r} = \hat{r}(\alpha, \beta, x) = (b + P_0\beta) - (A + E)x = b - (Ax + X\alpha) + P_0\beta = r + P_0\beta.$$

In ideal situations, we can impose the requirement that $\hat{r} = 0$ since $P_0\beta$ can play the role of the residual vector $r = \hat{r} - P_0\beta = b - (A + E)x$. However, this may not always be possible, due to the special structure that is imposed on E and $P_0\beta$.

When the structures imposed on E and $P_0\beta$ are such that $\hat{r}$ can be zero, the STLN solution that preserves the structure in b can be stated as

$$\min_{\hat{r}=0,\alpha,\beta,x} \left\| \begin{pmatrix} \hat{r}(\alpha, \beta, x) \\ D_1\alpha \\ D_2\beta \end{pmatrix} \right\|_p \tag{6.2}$$

for some diagonal matrices D_1 and D_2. This constrained minimization problem can be restated in different ways. We use the weighting method for the equality constrained

Algorithm STLNB

Input – A Structured Total Least Norm problem (1.2), with specified matrices A, D, P_0, vector b, and tolerance ϵ.

Output – Error matrix E and error vector β with the given affine structure in $[E \mid P_0\beta]$, vector x, and STLN error.

Begin

1. Choose a large number ω
   Set $E = 0$, $\alpha = 0$, $\beta = 0$, compute x from (2.5) and X from x, and set $\hat{r} = b - Ax$.
2. **repeat**
   (a) $\underset{\Delta x, \Delta\alpha, \Delta\beta}{\text{minimize}} \left\| \begin{bmatrix} \omega X & -\omega P_0 & \omega(A+E) \\ D_1 & 0 & 0 \\ 0 & D_2 & 0 \end{bmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta\beta \\ \Delta x \end{pmatrix} + \begin{pmatrix} -\omega\hat{r} \\ D_1\alpha \\ D_2\beta \end{pmatrix} \right\|_p .$
   (b) Set $x := x + \Delta x$, $\alpha := \alpha + \Delta\alpha$, $\beta := \beta + \Delta\beta$.
   (c) Construct E from α, and X from x. Compute $\hat{r} = (b + P_0\beta) - (A+E)x$.
   until ($\|\Delta x\|, \|\Delta\alpha\|, \|\Delta\beta\| \le \epsilon$)

End

least squares problems which transforms (6.2) into an unconstrained problem

$$\min_{\alpha,\beta,x} \left\| \begin{pmatrix} \omega\hat{r}(\alpha,\beta,x) \\ D_1\alpha \\ D_2\beta \end{pmatrix} \right\|_p \tag{6.3}$$

where ω is a large number [2, 16]. It can be shown that for $p = 2$ when ω approaches infinity, the solution for (6.3) converges to the solution for (6.2) when the constraint $\hat{r} = 0$ can be satisfied [2, 8, 16]. Thus, by using a large weight ω, we can obtain a good approximation for the solution for (6.2) by solving the unconstrained problem (6.3). For possible numerical problems associated with large ω, see [2, 16].

The algorithm is summarized in Algorithm STLNB. When there is no structure imposed on b, we can simply choose $\beta \in \Re^{m\times 1}$ to represent the perturbation on all the elements of b, and $P_0 = I$ accordingly. Therefore, Algorithm STLNB can handle the problems that can be solved by Algorithm STLN, although Algorithm STLN will be more efficient when there is no structure on b.

For the linear prediction, where $[A \mid b]$ or $[b \mid A]$ is Toeplitz and all the diagonals are subject to error, it is always possible to find a Toeplitz perturbation $[E \mid P_0\beta]$ such that

$$b + P_0\beta \in Range(A + E).$$

Therefore, $\hat{r}$ will become zero when the solution is obtained. Also, we can reformulate (6.2) into (6.3).

We will discuss the backward prediction only since the same results hold with forward prediction as well. When we need to impose Toeplitz structure on $[A \mid b]$, Step 2a of Algorithm STLNB can be further simplified since perturbation in b can be represented using the perturbation in A except for its first component. Specifically, if E is a Toeplitz matrix with its first column $[\alpha_n \cdots \alpha_{n+m-1}]^T$ and its first row $[\alpha_n \cdots \alpha_2\ \alpha_1]$, i.e.,

$$E = Toeplitz([\alpha_n \cdots \alpha_{n+m-1}]^T, [\alpha_n \cdots \alpha_2\ \alpha_1]) \text{ with } \alpha = [\alpha_1\ \alpha_2 \cdots \alpha_n \cdots \alpha_{n+m-1}]^T,$$
$$P_0 = I \qquad \text{with} \qquad \beta = [\beta_1\ \beta_2 \cdots \beta_m]^T,$$

then since $\beta_i = \alpha_{i-1}$, $i = 2, \cdots, m$, we have

$$\beta = \beta_1 \hat{e} + \hat{P}_0 \alpha \tag{6.4}$$

where

$$\hat{P}_0 = \begin{pmatrix} 0_{1\times(m-1)} & 0_{1\times n} \\ I_{(m-1)\times(m-1)} & 0_{(m-1)\times n} \end{pmatrix} \in \Re^{m\times(m+n-1)} \text{ and } \hat{e} = (\,1 \quad 0 \quad \cdots \quad 0\,)^T \in \Re^{m\times 1}.$$

From (6.4) and

$$X\Delta\alpha - P_0\Delta\beta + (A+E)\Delta x = (X - P_0)\Delta\alpha + (A+E)\Delta x - \Delta\beta_1\hat{e},$$

Step 2a of Algorithm STLNB is simplified to

$$\underset{\Delta\alpha,\Delta\beta_1,\Delta x}{\text{minimize}} \left\| \begin{pmatrix} \omega(X - \hat{P}_0) & -\omega\hat{e} & \omega(A+E) \\ D & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta\beta_1 \\ \Delta x \end{pmatrix} + \begin{pmatrix} -\omega\hat{r}(\alpha,\beta_1,x) \\ D\alpha \\ \beta_1 \end{pmatrix} \right\|_p \tag{6.5}$$

where $D^2 = diag(2, 3, \cdots, n+1, n+1, \cdots, 3, 2, 1) \in \Re^{(m+n-1)\times(m+n-1)}$, when all $m+n-1$ diagonals of A are different and subject to error. Then in Step 2b, β is modified so that $\beta_i = \alpha_{i-1}$ for $i = 2, \cdots, m$ and $\beta_1 := \beta_1 + \Delta\beta_1$.

7. Computational Results. The STLN algorithm has been implemented in MATLAB in order to investigate its computational performance. Each program for a different norm is denoted with the suffix p as STLNp with $p = 1, 2, \infty$. The computational testing has included over 200 relatively small problems with $m \leq 25$ and $n \leq 21$. In all cases, A has full rank. These computational tests represent a preliminary study of the effect on the algorithm behavior of structure, initial choice of x, and the magnitude of the minimum norm.

7.1. Convergence of STLN Algorithm. The numerical results obtained were consistent in showing that for each problem the STLN algorithm converged rapidly to a minimum solution for the problem. Since the function being minimized (as given by (1.2)) is not convex, there is no guarantee of convergence to a global minimum (for $p = 2$, convergence to a local minimum is discussed in Section 4). Typically, the STLN algorithm starts with x as given by (2.5). Other initial values of x were also used in some cases, in order to test the convergence. In every such case, the algorithm converged to a minimum value which was independent of the initial x. As a further confirmation, the Hessian matrix $H(y)$, as given by (4.5), was computed when the algorithm terminated, and was always found to be positive definite.

The convergence rate appears to be independent of problem size (over the range studied), but does depend on the size of the minimum norm. Specifically, a smaller minimum norm results in faster convergence. To illustrate the convergence, the results for two different problems will be summarized. These will be denoted as Problem I and Problem II which are defined as follows:

Problem I.

$m = 6$, $n = 4$, $q = 4$.

Matrix $A = $ Toeplitz(col, row) :

col $= [\ -3\quad 7\quad 10\quad -1\quad 0\quad 0\ ]^T$, row $= [\ -3\quad 0\quad 0\quad 0\ ]$,

Matrix $E = $ Toeplitz(col, row) :

col $= [\ \alpha_1\quad \alpha_2\quad \alpha_3\quad \alpha_4\quad 0\quad 0\ ]^T$, row $= [\ \alpha_1\quad 0\quad 0\quad 0\ ]$,

Two values of b were used:

$b^{(1)} = [\ -12\quad 25\quad 62\quad -59\quad 16\quad 100\ ]^T$,

$b^{(2)} = [\ -12\quad 25\quad 62\quad -59\quad 9\quad 122\ ]^T$.

Problem II.

$m = 9$, $n = 6$, $q = 4$.

Matrix $A = $ Toeplitz(col, row):

col $= [\ 3\quad -1\quad -6\quad 2\quad 5\quad 0\quad -8\quad -7\quad 1\ ]^T$,

row $= [\ 3\quad -6\quad 2\quad 0\quad 8\quad -4\ ]$,

Matrix $E = $ Toeplitz(col, row):

col $= [\ \alpha_3\quad \alpha_4\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\ ]^T$,

row $= [\ \alpha_3\quad \alpha_2\quad \alpha_1\quad 0\quad 0\quad 0\ ]$,

$b = [\ 62\quad 62\quad 5\quad 22\quad -65\quad -60\quad -10\quad 86\quad 101\ ]^T$.

Note that in Problem I the matrix E has the same nonzero diagonal patterns as A, whereas in Problem II, E has only 4 nonzero diagonals. This means that in Problem II, only those 4 diagonals of the Toeplitz matrix $A + E$ can change. Also as shown in Table 7.1, the vector $b^{(1)}$ is more closely approximated by the columns of A, than is the case for $b^{(2)}$.

To illustrate the structure of the matrices $P_i, i = 1, \ldots, m$, the matrices P_1 and P_2 for Problem II are now given:

$$P_1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \text{and} \quad P_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The following solutions were obtained for these problems:

- Least Squares (LS),
- Total Least Squares (TLS),
- Structured Total Least Norm (STLN), for $p = 2$, and $p = 1, \infty$.

The LS and TLS solutions were obtained with MATLAB using the QR decomposition and the SVD, respectively. The STLN solutions were obtained using Algorithm STLN given in Section 2.

For $p = 2$, Step 2a of the algorithm computes a LS solution. For $p = \infty$, Step 2a is essentially the solution of the linear program (5.2) in Section 5. The computed results for Problem I, using these different approaches, are summarized in Fig. 7.1 and 7.2, and Table 7.1.

Fig. 7.1(a) shows the convergence of the STLN algorithm to the minimum value of the Total Norm (TN), as given by (1.2). This minimum value computed from the STLN algorithm is the STLN. Fig. 7.1(a) shows the value of the TNerr at each iteration of the STLN algorithm, where

$$\text{TNerr} = \text{TN} - \text{STLN}. \tag{7.1}$$

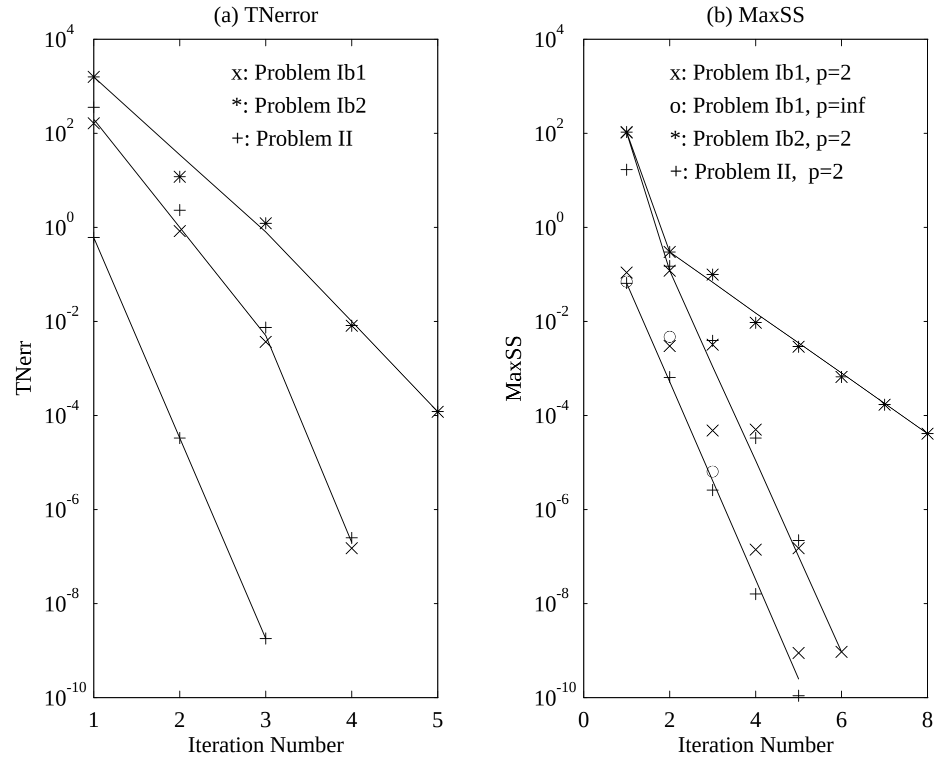

FIG. 7.1. *(a) Convergence of Total Norm to STLN with $p = 2$ (b) Convergence of step size*

The convergence (using $p = 2$) is shown for 4 different cases: Problem Ib_1 (Problem I, with $b = b^{(1)}$), Problem Ib_2 (Problem I, with $b = b^{(2)}$), and Problem II with two different initial values of x. The initial value $\alpha = 0$ was used for all cases. These results show convergence in 3 iterations to TNerr $\leq 5 \times 10^{-7}$, for both Problems Ib_1 and II, even when the initial TNerr is very large (~ 100). The large initial value of TNerr is obtained by using an initial value of x very different from its converged value. The smaller initial TNerr shown for Problem II, was obtained by using $x = x_{ls}$ as the initial value, where x_{ls} is the least squares solution.

Convergence for Problem Ib_2 is seen to be significantly slower, with 4 iterations needed to obtain TNerr $\cong 10^{-4}$, from an initial TNerr $\cong 10^3$. Essentially the same convergence rate was obtained for Problem Ib_2, starting with an initial TNerr $\cong 10$, obtained with $x = x_{ls}$. To avoid complicating the figure, this last case is not included in Fig. 7.1(a).

In order to understand these convergence results, it is important to note that both the STLN and the minimum residual norm are at least 10 times greater for Problem Ib_2 than they are for Problem Ib_1. These values are given in Table 7.1, to be discussed shortly.

A closely related aspect of the convergence of the STLN algorithm is given in Fig. 7.1(b), which shows the rate of decrease of the step size with iteration number. For purposes of this graph the Maximum Step Size (MaxSS) is defined as

$$\text{MaxSS} = \max\{\|\Delta\alpha\|_\infty, \|\Delta x\|_\infty\}, \tag{7.2}$$

$$E_{TLS} = \begin{pmatrix} -4.5\times10^{-3} & -7.7\times10^{-3} & 4.0\times10^{-6} & 7.9\times10^{-3} & 9.1\times10^{-3} & 9.1\times10^{-3} \\ -1.4\times10^{-3} & -2.4\times10^{-3} & 1.2\times10^{-6} & 2.4\times10^{-3} & 2.8\times10^{-3} & 2.8\times10^{-3} \\ 3.7\times10^{-3} & 6.4\times10^{-3} & -3.3\times10^{-6} & -6.5\times10^{-3} & -7.5\times10^{-3} & -7.5\times10^{-3} \\ -1.5\times10^{-4} & -2.6\times10^{-4} & 1.4\times10^{-7} & 2.7\times10^{-4} & 3.1\times10^{-4} & 3.1\times10^{-4} \\ -2.2\times10^{-3} & -3.8\times10^{-3} & 2.0\times10^{-6} & 3.9\times10^{-3} & 4.5\times10^{-3} & 4.5\times10^{-3} \\ -5.5\times10^{-3} & -9.5\times10^{-3} & 4.9\times10^{-6} & 9.6\times10^{-3} & 1.2\times10^{-2} & 1.1\times10^{-2} \\ -6.8\times10^{-3} & -1.2\times10^{-2} & 6.0\times10^{-6} & 1.2\times10^{-2} & 1.4\times10^{-2} & 1.4\times10^{-2} \\ 8.0\times10^{-4} & 1.4\times10^{-3} & -7.1\times10^{-7} & -1.4\times10^{-3} & -1.6\times10^{-3} & -1.6\times10^{-3} \\ -2.6\times10^{-3} & -4.5\times10^{-3} & 2.3\times10^{-6} & 4.5\times10^{-3} & 5.2\times10^{-3} & 5.2\times10^{-3} \end{pmatrix}$$

$$E_{STLN2} = \begin{pmatrix} -2.6\times10^{-2} & -1.6\times10^{-3} & 2.0\times10^{-2} & 0 & 0 & 0 \\ 6.5\times10^{-2} & -2.6\times10^{-2} & -1.6\times10^{-3} & 2.0\times10^{-2} & 0 & 0 \\ 0 & 6.5\times10^{-2} & -2.6\times10^{-2} & -1.6\times10^{-3} & 2.0\times10^{-2} & 0 \\ 0 & 0 & 6.5\times10^{-2} & -2.6\times10^{-2} & -1.6\times10^{-3} & 2.0\times10^{-2} \\ 0 & 0 & 0 & 6.5\times10^{-2} & -2.6\times10^{-2} & -1.6\times10^{-3} \\ 0 & 0 & 0 & 0 & 6.5\times10^{-2} & -2.6\times10^{-2} \\ 0 & 0 & 0 & 0 & 0 & 6.5\times10^{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

FIG. 7.2. *E matrices obtained by TLS and STLN2 for Problem II.*

and is shown as a function of the iteration number. This is a more sensitive measure of convergence, since small changes in α and x may continue even when TNerr is very small. This is most likely to occur when the minimum of TN is very flat.

The results for a total of 6 cases are presented in Fig. 7.1(b). These include the 4 cases shown in Fig. 7.1(a), and in addition, Problem Ib_1, with the initial value of $x = x_{ls}$, and Problem Ib_1, using the L_∞ norm with an initial value of x given by (2.5) with $p = \infty$. The results shown in Fig. 7.1 are typical of all the problems solved by the STLN algorithm. For all the problems we tested, the STLN algorithm converges to the global minimum from the chosen initial value of x, and the convergence rate is independent of the norm used (note that the data for Problems Ib_1 ($p = 2$ and $p = \infty$) and II, all lie on the lowest curve of Fig. 7.1(b)). The convergence rate is second order for small residual problems, and apparently superlinear for larger residual problems (compare Problems Ib_1 and Ib_2). The minimum norm for all problems tested (except Ib_2) was similar to that in Ib_1 and II, and all these problems converged in, at most, 6 iterations.

These computational results are consistent with the analysis given in Section 4. The dependence of the convergence rate on the residual minimum norm is clearly shown by reference to Table 7.1, to be discussed below. A more complete understanding of this dependence requires further investigation, both theoretical and computational.

7.2. Comparison of STLN with LS and TLS. A direct comparison of the STLN ($p = 2$) solution with the TLS solution, for Problem II, is shown in Fig. 7.2. For this problem the matrix E is Toeplitz, with only four non-zero diagonals. The computed matrix E is shown for the TLS solution and the STLN solution. As expected for the TLS solution, all elements of E are non-zero, and it does not have a Toeplitz structure. That is, the TLS solution allows all elements of the matrix A to change. This is in contrast to the STLN solution where only the 4 designated diagonals are allowed to change, so that $A + E$ preserves the original Toeplitz structure of A.

	$b^{(1)}$				$b^{(2)}$			
	LS	TLS	STLN2	STLN∞	LS	TLS	STLN2	STLN∞
rnorm	8.23×10^{-1}	6.14×10^{-3}	2.20×10^{-2}	0	10.445	5.72×10^{-2}	5.359×10^{-1}	0
Enorm	0	7.08×10^{-2}	1.09×10^{-1}	7.24×10^{-2}	0	7.72×10^{-1}	1.432	1.136
Tnorm	8.23×10^{-1}	7.11×10^{-2}	1.11×10^{-1}	7.24×10^{-2}	10.445	7.74×10^{-1}	1.529	1.136
x_1	4.0292	4.0292	3.9638	3.9652	3.4739	3.4650	4.3948	4.2865
x_2	0.9056	0.9058	1.0090	1.0058	1.7889	1.8252	0.2927	0.0489
x_3	-5.0122	-5.0126	-5.1025	-5.1289	-6.3357	-6.3864	-5.0594	-4.994
x_4	9.5310	9.5314	9.5596	9.5937	11.157	11.221	10.924	11.017

TABLE 7.1
Minimum norms and x for Problem I.

Finally, the computed norms for Problems Ib_1, Ib_2, and the corresponding x vectors are given in Table 7.1. This table compares the minimum norm solutions obtained by least squares (LS), where $E = 0$; by TLS, where all elements of E can change; and by the STLN algorithm (for both $p = 2$ and $p = \infty$), where only the specified elements of E can change. For each case, the following three norms are tabulated:

$$\begin{aligned} \text{rnorm} &= \|r\|_p, \\ \text{Enorm} &= \|D\alpha\|_p, \quad \text{and} \\ \text{Tnorm} &= \left\| \begin{pmatrix} r \\ D\alpha \end{pmatrix} \right\|_p \end{aligned} \tag{7.3}$$

It should be noted that for each problem the Tnorm satisfies the inequality:

$$\text{TLS} \le \text{STLN2} \le \text{LS} \tag{7.4}$$

This is the expected result, since TLS is unconstrained (all elements of E can change), STLN is partially constrained (only specified elements of E can change), and LS is completely constrained ($E = 0$). Also, the minimum residual for Problem Ib_2 is at least 10 times greater than it is for Problems Ib_1 and II, which seems to be the significant property affecting the convergence rate.

In addition to the computational convergence of the STLN algorithm, the properties of the solution obtained were investigated. In particular, the vector x and error matrix E obtained were compared for LS, TLS, and STLNp, for $p = 1, 2, \infty$. There are a number of ways in which this comparison can be made. The comparison used is based on the assumption that there exists a 'correct' structured matrix A_c and vector b_c, such that

$$A_c x_c = b_c \tag{7.5}$$

for some 'correct' vector x_c. In other words, error-free values exist such that the overdetermined system has a solution x_c, with zero residual. The actual data contains noise so that a perturbed (but structure preserving) matrix A_p, and vector b_p are known. The objective is to get the "best" solution x_p, to the perturbed system $A_p x \approx b_p$ and, to the extent possible, reconstruct the matrix A_c and vector b_c from the noisy data. Specifically, the error matrix E, and residual vector r, are computed so that:

$$(A_p + E)x_p = b_p - r \tag{7.6}$$

This is done by minimizing the appropriate norm of E and r.

Method	b_{err}	A_{err}	x_{err}
LS	3.3e-4	6.5e-4	1.8e-3
TLS	2.4e-5	6.5e-4	1.8e-3
STLN2	2.2e-5	9.7e-5	2.0e-4
STLN1	2.4e-5	7.6e-5	2.1e-4
STLN∞	2.6e-5	1.6e-4	3.4e-4

TABLE 7.2
Solution Accuracy, A and b unstructured, $m = 20,\ n = 16,\ q = 4$.

Method	b_{err}	A_{err}	x_{err}
LS	8.9e-3	1.2e-2	1.5e-2
TLS	5.3e-4	1.2e-2	1.5e-2
STLN2	4.3e-4	5.3e-4	7.3e-4
STLN1	5.5e-4	5.4e-4	3.5e-4
STLN∞	5.7e-4	5.2e-4	5.9e-4

TABLE 7.3
Solution Accuracy, A Toeplitz, b unstructured, $m = 11,\ n = 6,\ q = 4$

Method	A_{err}	x_{err}
LS	4.4e-3	1.4e-1
TLS	4.0e-3	2.4e-2
STLN2	3.8e-3	3.3e-3
STLN1	2.2e-6	7.2e-6
STLN∞	4.7e-3	1.3e-1

TABLE 7.4
Solution Accuracy, $[A \mid b]$ *Toeplitz,* $m = 14,\ n = 4,\ q = 17$

The test problems are constructed so that A_c, b_c, and x_c are known. Then random perturbations are generated to give A_p and b_p, so that A_p and b_p preserve the same structure as A_c and b_c. The matrix E and r, x_p, satisfying (7.6) are then computed via LS, TLS, and STLN.

A comparison of these errors for LS, TLS, and STLN was made for three different types of structured problems:

1. The matrix A and vector b are unstructured, but errors can occur only in certain elements.
2. The matrix A is Toeplitz, with b unstructured.
3. The matrix $[A \mid b]$ is Toeplitz.

To illustrate the comparison obtained with over 200 test problems, a typical case has been selected for each type of structured problem. These typical cases are presented in Tables 7.2, 7.3, and 7.4. The following quantities are tabulated to give the measure of robustness [8]:

$$\begin{aligned} b_{pert} &= ||b_p - b_c||_2/||b_c||_2 \\ A_{pert} &= ||A_p - A_c||_F/||A_c||_F \\ b_{err} &= ||b_p - r - b_c||_2/||b_c||_2 \\ A_{err} &= ||A_p + E - A_c||_F/||A_c||_F \\ x_{err} &= ||x_p - x_c||_2/||x_c||_2 \end{aligned}$$

Table 7.2 gives the comparison for A and b unstructured, with all elements of

b and 4 elements of A perturbed, and $m = 20$, $n = 16$, and $q = 4$. The values of $b_{pert} = 2.4e{-}5$ and $A_{pert} = 6.5e{-}4$, were used. The matrices A_c and A_p are dense, but E is sparse with only 4 nonzero elements, for the STLN solutions. The matrix E is zero for LS and dense for TLS.

Table 7.3 gives the comparison for A Toeplitz, and b unstructured, with $m = 11$, $n = 6$, and $q = 4$. The matrices A_c and A_p are both Toeplitz, and E is Toeplitz with 4 nonzero diagonals for the STLN solutions. The matrix E is zero for LS and dense for TLS. The values of $b_{pert} = 5.2e{-}4$, and $A_{pert} = 1.2e{-}2$, were used.

Table 7.4 gives the comparison for $[A \mid b]$ Toeplitz, with $m = 14$, $n = 4$ and $q = 17$. The problem presented was selected to illustrate the performance of the STLN algorithm with an outlier in the data. In addition to small random perturbations in each diagonal of $[A \mid b]$, a much larger error was introduced in one of the diagonals. The small random perturbations gave $A_{pert} = 2.1e{-}6$ and the exact data with outlier only gave $A_{pert} = 5.1e{-}3$. The b_{err} is included in A_{err} for $[A \mid b]$ Toeplitz. The matrix E is zero for LS and dense for TLS. The most significant result shown in Table 7.4 is that STLN1 is essentially unaffected by the outlier. Note that the STLN∞ solution is affected most by the outlier.

The results presented in Tables 7.2, 7.3, and 7.4 show that the errors in the STLN1 and STLN2 solutions are significantly less than in the LS and TLS solutions. Similar results were obtained for all structured problems of these types tested.

8. Conclusions and Future Work. A new algorithm has been presented for solving an important class of problems related to Total Least Squares. The main new features of this approach are that it preserves the problem structure, and also permits the minimization of error in different norms. Both the theoretical analysis and the computational results show that the STLN algorithm is an efficient computational method for problems with a special structure, or where the number of elements with possible error (in the matrix A) is not too large. The ability to minimize the error in norms other than the 2-norm is also important, since we believe this will give more robust solutions in certain cases. When the data are from the complex field, the presented algorithms can be used in a straight forward way for the 2-norm. For the 1-norm and ∞-norm with complex data, the STLN problem will require the solution of a nonlinear programming problem rather than linear programming.

In order to more fully investigate the potential of the STLN formulation and algorithm for a range of applications, a number of areas need further study. Future work on STLN will include computational testing of much larger problems arising in important applications where the matrix A has a special, or sparse structure, and theoretical and computational analysis of the effect on the convergence rate of the magnitude of the minimum norm.

Acknowledgement. We wish to thank Dr. Sabine Van Huffel, for introducing us to this problem and subsequent discussions, and Prof. Philip Gill for a helpful discussion.

REFERENCES

[1] T.J. Abatzoglou, J.M. Mendel, G.A. Harada. The constrained total least squares technique and its application to harmonic superresolution. *IEEE Transactions on Signal Processing*, 39:1070–1087, 1991.

[2] A.A. Anda and H. Park. Self-scaling fast rotations for stiff least squares problems, to appear in *Linear Algebra and its Applications.*

[3] J. W. Demmel. The smallest perturbation of a submatrix which lowers the rank and constrained total least squares problems. *SIAM Journal on Numerical Analysis*, 24:199–206, 1987.

[4] B. De Moor. Structured total least squares and L_2 approximation problems. *Linear Algebra and its Applications*, special issue on Numerical Linear Algebra Methods in Control, Signals and Systems, Eds. Van Dooren, et. al., 188-189:163-207, 1993.

[5] R. Fletcher. *Practical Methods of Optimization, Vol. 1, Unconstrained Optimization.* John Wiley, New York, 1980.

[6] E. Fogel. Total least squares for Toeplitz structures. In proceedings of the 21st IEEE Conference on Decision and Control, Orlando, 3:1003–1004, 1982.

[7] G. H. Golub and C. F. Van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17:883–893, 1980.

[8] G. H. Golub and C. F. Van Loan. *Matrix Computations*, second edition. Johns Hopkins University Press, Baltimore, 1989.

[9] S. Haykin. *Adaptive Filter Theory*, second edition. Prentice Hall, Englewood Cliffs, NJ, 1991.

[10] M. R. Osborne and G. A. Watson. An analysis of the total approximation problem in separable norms and an algorithm for the total l_1 problem. *SIAM J. Sci. Stat. Comput.*, 6:410–424, 1985.

[11] M. A. Rahman and K. B. Yu. Total least squares approach for frequency estimation using linear prediction. *IEEE Transactions on Acoustic Speech Signal Processing*, ASSP–35:1440–1454, 1987.

[12] J.B. Rosen H. Park, and J. Glick. Total least norm formulation and solution for structured problems. AHPCRC preprint 94-041, University of Minnesota, July, 1994.

[13] S. Van Huffel and J. Vandewalle. Analysis and properties of the generalized total least squares problem $AX \approx B$ when some or all columns of A are subject to errors. *SIAM Journal on Matrix Anal. Appl.*, 10:294–315, 1989.

[14] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem, Computational Aspects and Analysis.* SIAM, Philadelphia, 1991.

[15] S. Van Huffel, B. De Moor, and H. Chen. Relationships between constrained and structured TLS, with applications to signal enhancement. *Proceedings of the Int. Symposium on the Mathematical Theory for Networks and Systems*, Regensburg, Germany, August 2-6, 1993, to appear.

[16] C.F. Van Loan. On the method of weighting for equality constrained least squares problems. *SIAM J. Numer. Anal.*, 22:851–864, 1985.

[17] G.A. Watson. On a class of algorithms for total approximation. *Journal of Approximation Theory*, 45: 219-231, 1985.