SIAM J. MATRIX ANAL. APPL.
Vol. 20, No. 1, pp. 14–30

STRUCTURED TOTAL LEAST NORM FOR NONLINEAR PROBLEMS*

J. B. ROSEN†, HAESUN PARK‡, AND JOHN GLICK§

Abstract. An extension of the recently developed structured total least norm (STLN) problem formulation is described for solving a class of nonlinear parameter estimation problems. STLN is a problem formulation for obtaining an approximate solution to the overdetermined linear system $Ax \approx b$ preserving the given affine structure in A or $[A \mid b]$, where errors can occur in both the vector b and the matrix A. The approximate solution can be obtained to minimize the error in the L_p norm, where $p = 1, 2,$ or ∞. In the extension of STLN to nonlinear problems, the elements of A may be differentiable nonlinear functions of a parameter vector, whose value needs to be approximated. We call this extension structured nonlinear total least norm (SNTLN). The SNTLN problem is formulated and its solution by a modified STLN algorithm is described. Optimality conditions and convergence for the 2-norm case are presented.

Computational tests were carried out on an overdetermined system with Vandermonde structure and on two nonlinear parameter estimation problems. In these problems, both the coefficients and the unknown parameters were to be determined. The computational results demonstrate that the SNTLN algorithm recovers good approximations to the correct values of both the coefficients and parameters, in the presence of noise in the data and poor initial estimates of the parameters. It is also shown that the SNTLN algorithm with the 1-norm minimization is robust with respect to outliers in the data.

Key words. overdetermined linear systems, data fitting, least squares, linear prediction, parameter estimation, frequency estimation, outliers, total least norm, total least squares, Vandermonde matrix, 1-norm, 2-norm

AMS subject classifications. 65F20, 65F30, 65M10, 65Y20

PII. S0895479896301662

1. Introduction.

A new algorithm called structured total least norm (STLN) has recently been developed [24] for obtaining an approximate solution to the overdetermined linear system

$$Ax \approx b,$$

where errors may occur in both the vector b and in elements of the affinely structured $(m \times n)$ matrix A, where $m > n$. The STLN algorithm preserves the structure of A or $[A \mid b]$ and minimizes, in a suitable norm, the residual and the change in the elements of A. This minimization can be done using the L_p norm, where $p = 1, 2,$ or ∞.

*Received by the editors April 8, 1996; accepted for publication (in revised form) by S. Van Huffel June 20, 1997; published electronically September 15, 1998. The work of the first and third authors was supported in part by Air Force Office of Scientific Research grant AFOSR-91-0147 and the Minnesota Supercomputer Institute. The work of the first author was supported in part by ARPA/NIST grant 60NANB4D1615 and National Science Foundation grant CCR-9509085. The work of the second author was supported in part by National Science Foundation grants CCR-9209726 and CCR-9509085.

http://www.siam.org/journals/simax/20-1/30166.html

†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 (rosen@cs.umn.edu, jbrosen@cs.ucsd.edu).

‡Computer Science Department, University of Minnesota, Minneapolis, MN 55455 (hpark@cs.umn.edu).

§Department of Mathematics and Computer Science, University of San Diego, San Diego, CA 92110 (glick@pwa.acusd.edu).

The STLN algorithm addresses the same class of problems as total least squares [30], but has the advantage that it preserves the structure of A or $[A \mid b]$ when they are perturbed in the solution. The theory, implementation, and some applications of the STLN algorithm are presented in several recent papers [24, 25, 21, 28, 20, 7].

In this paper, we show that the STLN algorithm can be extended to solve a related, but more difficult, structured approximation problem, in which the elements of A may be *nonlinear* differentiable functions of a parameter vector α. We call this extension structured nonlinear total least norm (SNTLN). The SNTLN algorithm will be described, and the optimality conditions and convergence properties of the algorithm are also given for the L_2 norm. Its relationship to the Gauss–Newton method is shown. The details of the SNTLN algorithm for the Vandermonde structured problems are shown. Some computational tests of the algorithm, using the L_1 and L_2 norms are summarized. These tests were carried out on overdetermined systems for which an exact solution with zero residual is known. That is, an exact parameter α_c, the right-hand side vector b_c, and the exact solution x_c are known so that

$$A(\alpha_c)x_c = b_c.$$

In typical applications, there is error in the vector b, and the parameter vector α_c is not known but is to be estimated. In the SNTLN algorithm, an initial estimate $\hat{\alpha}$ is needed to start the iterative solution. In some applications, an initial value $\hat{\alpha}$ is either given or can be estimated. In this case, the SNTLN algorithm is like the STLN algorithm in the sense that it solves the given overdetermined system preserving the given problem structure. The difference is that the SNTLN preserves nonlinear structures. In fact, when the SNTLN algorithm is applied to affinely structured problems, it becomes the STLN algorithm.

The computational results show that for both L_1 and L_2, the algorithm converged from a range of initial values $\hat{\alpha}$, to α_c for $b = b_c$, and to a value close to α_c for $\|b - b_c\|$ small. Furthermore, the robust behavior with respect to outliers in the vector b was demonstrated by the SNTLN algorithm using the L_1 norm. Such robust behavior has previously been observed by Barrodale and others [23, 4, 5, 17] for the simpler problem of finding an x such that the residual norm $\|b - Ax\|_1$ is minimized. The use of the L_1 norm for some nonlinear problems has also been investigated in [27, 31].

Before describing SNTLN we briefly summarize the earlier STLN formulation and algorithm. The formulation takes full advantage of the special structure of the given matrix A. In particular, when there are $q \leq mn$ elements of A which are subject to error, a $q \times 1$ vector α is used to represent the corresponding elements of the error matrix E, which gives

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

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 or Hankel matrix, each diagonal or antidiagonal consists of elements with the same value, respectively, so $q \leq m + n - 1$.

The matrix E is specified by those elements of A which may be subject to error. Each different nonzero element of E corresponds to one of the α_k, $k = 1, \ldots, q$. Now, the residual vector $r = b - (A + E)x$ is a function of (α, x). Let $D \in \mathbf{R}^{q \times q}$ be a diagonal matrix that accounts for the repetition of elements of α in the matrix E. Then the STLN problem can be stated as follows.

ALGORITHM STLN.
Input – A Structured Total Least Norm problem (1.1), with matrices A, D, vector b, and tolerance *tol*
Output – Error matrix E, residual vector r, and vector x

1. Set $\alpha = 0$, $E = 0$, compute x from (1.4), construct X from x, and set $r = b - Ax$.
2. **repeat**
   (a) $\underset{\Delta x, \Delta\alpha}{\text{minimize}} \left\| \begin{pmatrix} X & A+E \\ D & 0 \end{pmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} + \begin{pmatrix} -r \\ D\alpha \end{pmatrix} \right\|_p$.
   (b) Set $x := x + \Delta x$, $\alpha := \alpha + \Delta\alpha$.
   (c) Construct X from x, and E from α. Compute $r = b - (A+E)x$.
   until ($\|\Delta x\| \le tol$ and $\|\Delta\alpha\| \le tol$)

$$\min_{\alpha, x} \left\| \begin{matrix} r(\alpha, x) \\ D\alpha \end{matrix} \right\|_p, \tag{1.1}$$

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

In the iterative algorithm for solving the STLN problem, the vector Ex is represented in terms of α. This is accomplished by defining an $m \times q$ matrix X such that $X\alpha = Ex$. The matrix X consists of the elements of x, with suitable repetition, giving X a special structure. If E is Toeplitz and every diagonal in A is subject to error, then X can be arranged to be Toeplitz too. In fact, E and X have exactly the same number of nonzero elements.

In the minimization (1.1), a linear approximation to $r(\alpha, x)$ is used. Let Δx represent a small change in x, and ΔE a small change in the variable elements of E. Then we have $X(\Delta\alpha) = (\Delta E)x$, where $\Delta\alpha$ represents the corresponding small change in the elements of α. Neglecting the second-order terms in $\|\Delta\alpha\|$ and $\|\Delta x\|$,

$$r(\alpha + \Delta\alpha, x + \Delta x) = r(\alpha, x) - X\Delta\alpha - (A+E)\Delta x. \tag{1.2}$$

The linearization of (1.1) now becomes

$$\min_{\Delta\alpha, \Delta x} \left\| \begin{pmatrix} X & A+E \\ D & 0 \end{pmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} + \begin{pmatrix} -r \\ D\alpha \end{pmatrix} \right\|_p. \tag{1.3}$$

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

$$\min_x \|b - Ax\|_p. \tag{1.4}$$

The STLN algorithm is summarized in Algorithm STLN.

2. SNTLN. We now show how the nonlinear extension is formulated, and how Algorithm STLN can be modified to solve this nonlinear problem. We describe the extension to the case in which the matrix A is a differentiable function $A(\alpha)$ of an $s \times 1$ parameter vector α and an approximate solution to the overdetermined system

$$A(\alpha)x \approx b \tag{2.1}$$

is to be obtained. Any number of the elements of A may depend on α, but important properties of the solution will depend on m, n, and s. This is discussed in section 3. The residual vector $r = r(\alpha, x)$ is now defined by

$$r(\alpha, x) = b - A(\alpha)x \tag{2.2}$$

and the parametric problem can be stated as

$$\min_{\alpha,x} \left\| \begin{array}{c} r(\alpha, x) \\ D(\alpha - \hat{\alpha}) \end{array} \right\|_p, \tag{2.3}$$

where $\hat{\alpha}$ is an initial estimate of the optimum parameter vector and D is a diagonal matrix of positive weights. For the assumptions on rank of A and Hessians of the elements of A, see section 3.

The nonlinear parameter estimation problem of minimizing the L_2-norm of $r(\alpha, x)$, where the problem is linear in x but nonlinear in α, has been investigated in many earlier papers [10, 12, 18, 6, 16]. It is often designated as "separable nonlinear least squares" [6, 10, 12, 18, 13]. The method of solution presented here, as an extension of the STLN algorithm and called the SNTLN algorithm, is different from those presented earlier.

The problem (2.3) clearly reduces to those considered earlier when $p = 2$ and $D = 0$. More importantly, the solution method given here, the SNTLN algorithm, applies directly to the problem (2.3) with $p = 1$ or ∞, in addition to $p = 2$. As shown in section 5, the use of $p = 1$ has important practical applications because of its robustness with respect to outliers in the data.

The most closely related of these earlier methods is variable projection [10, 12]. However, variable projection is only valid for $s \leq m - n$, whereas SNTLN, with positive D, is not similarly restricted. Furthermore, the SNTLN algorithm is valid for all three norms, $p = 1, 2, \infty$, while the variable projection method is limited to the L_2 norm. Starting with the same initial estimate $\hat{\alpha}$, the sequence of vectors $\{\alpha_k, x_k\}$ computed by SNTLN, with $p = 2$, will differ from those computed by the variable projection method. Specifically, variable projection x_k will always be the least squares solution of $A(\alpha_k)x \approx b$; this may not be true for SNTLN, except at termination. A complete statement on the increments $(\Delta\alpha, \Delta x)$ used in SNTLN is given in section 3.

The more general nonlinear model

$$\min_y \|f(y)\|_p,$$

$$f : \mathbf{C}^n \to \mathbf{C}^m, \quad m > n,$$

can also be solved by the SNTLN algorithm. With appropriate assumptions on $f(y)$, this is shown in [26]. In this paper, we limit consideration to the separable case

$$y = \begin{pmatrix} \alpha \\ x \end{pmatrix}, \qquad f = \begin{pmatrix} b - A(\alpha)x \\ D(\alpha - \hat{\alpha}) \end{pmatrix}.$$

Just as for the STLN algorithm, we compute the minimum solution to (2.3) iteratively by linearizing $r(\alpha, x)$:

$$r(\alpha + \Delta\alpha, x + \Delta x) = r(\alpha, x) - A(\alpha)\Delta x - J(\alpha, x)\Delta\alpha, \tag{2.4}$$

ALGORITHM SNTLN.
Input – Matrices $A(\alpha)$, $\nabla_\alpha a_i$, $1 \le i \le n$, D, vector b, initial estimate $\hat{\alpha}$, and tolerance tol.
Output – α, residual vector r, and vector x

1. Set $\alpha = \hat{\alpha}$, compute x from (1.4) with $A = A(\alpha)$, $J(\alpha, x)$, and set $r = b - A(\alpha)x$.
2. **repeat**
   (a) $\underset{\Delta x, \Delta\alpha}{\text{minimize}} \left\| \begin{pmatrix} A(\alpha) & J(\alpha,x) \\ 0 & D \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\alpha \end{pmatrix} + \begin{pmatrix} -r \\ D(\alpha - \hat{\alpha}) \end{pmatrix} \right\|_p$.
   (b) Set $x := x + \Delta x$, $\alpha := \alpha + \Delta\alpha$.
   (c) Compute $J(\alpha, x)$, $A(\alpha)$, $r = b - A(\alpha)x$.
   until ($\|\Delta x\| \le tol$ and $\|\Delta\alpha\| \le tol$)

where $J(\alpha, x)$ is the Jacobian, with respect to α, of $A(\alpha)x$. Let $a_j(\alpha)$ represent the jth column of $A(\alpha)$. Then

$$J(\alpha, x) = \nabla_\alpha (A(\alpha)x) = \sum_{j=1}^{n} x_j \nabla_\alpha a_j(\alpha). \tag{2.5}$$

The algorithm for the nonlinear extension is identical to the STLN algorithm as given above, except that now the matrix X is replaced by $J(\alpha, x)$, and $A + E$ is replaced by $A(\alpha)$, wherever it is appropriate. The modification of STLN for nonlinear parameter estimation is given by Algorithm SNTLN. For the formulation of step 2(a) of Algorithm SNTLN for $p = 1$ and $p = \infty$ as a linear program, see [24, 26].

It should be noted that STLN is a special case of SNTLN, and Algorithm SNTLN becomes Algorithm STLN for affinely structured problems. Specifically, for affinely structured problems, in Algorithm STLN, we have

$$A(\alpha) = A + E(\alpha),$$

where $\hat{\alpha} = 0$ and $E(0) = 0$ according to step 1, which gives

$$A(\hat{\alpha}) = A(0) = A + E(0) = A.$$

Therefore,

$$A(\alpha)x = Ax + E(\alpha)x = Ax + X\alpha$$

and

$$J(\alpha, x) = \nabla_\alpha [A(\alpha)x] = X.$$

Note that using a small D in Step 2(a) of Algorithm STLN or Algorithm SNTLN makes only a small change in the condition of the problem. This is due to the fact that

$$\operatorname{cond}\left(\begin{pmatrix} A & J \\ 0 & D \end{pmatrix} \right) \approx \operatorname{cond}((A \quad J))$$

when D is small. Also, the stable algorithms for computing the QR decomposition, such as those based on Givens and Householder transformations, are not sensitive to having small rows $(\, 0 \quad D \,)$ placed at the bottom of the matrix. For details, see [2].

Important applications of SNTLN include problems of estimating parameters. In some parameter estimation applications, it is assumed that the system, with no noise, can be represented by

$$y(t) = \sum_{j=1}^{n} x_j f_j(\alpha, t),$$

where the $f_j(\alpha, t)$ are specified functions of α and t. The functional dependence of the f_j on α is known, and it is desired to estimate the "best" values of α and x. Measurements of the system at $m > n + s$ points t_i, $i = 1, \ldots, m$, are taken, giving the $m \times 1$ vector b that represents y. Note that with the SNTLN algorithm presented here there is no requirement that the t_i be uniformly spaced.

This problem can immediately be put in the desired form by defining the elements of $A(\alpha)$ as

$$A(\alpha) = (a_{ij}) = f_j(\alpha, t_i),$$

which gives

$$\begin{pmatrix} f_1(\alpha,t_1) & f_2(\alpha,t_1) & \cdots & f_n(\alpha,t_1) \\ f_1(\alpha,t_2) & f_2(\alpha,t_2) & \cdots & f_n(\alpha,t_2) \\ \vdots & & & \vdots \\ f_1(\alpha,t_m) & f_2(\alpha,t_m) & \cdots & f_n(\alpha,t_m) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \approx \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}. \tag{2.6}$$

This type of problem arises in a variety of signal processing applications, such as frequency and exponential decay estimation [14, 22, 1]. In another potential application, the matrix $A(\alpha)$ is large and sparse, with only a few elements depending on α. An example of this kind of problem occurs in fitting scattered data in three-dimensional space [11].

Another type of application involves solving an overdetermined system

$$A(\hat{\alpha})x \approx b$$

for x, where A is a nonlinear function of a vector α and the matrix $A(\hat{\alpha})$ is given. In this case, the SNTLN can be used to solve the system while preserving the given *nonlinear* structure in $A(\alpha)$. The only difference is that here the initial value $\hat{\alpha}$, for α, is given.

3. Optimality conditions and convergence. In this section we generalize the STLN convergence results presented in [24] to the SNTLN algorithm. These results hold for the SNTLN algorithm using the L_2 norm, where the function being minimized is differentiable.

Some additional assumptions on $A(\alpha)$ are needed in order to obtain the convergence results. We assume that with no noise (or error) in $b = b_c$, there is a correct parameter vector α_c, and corresponding x_c, such that $A(\alpha_c)x_c = b_c$. A neighborhood Ω_c in the s-dimensional space that contains the initial estimate $\hat{\alpha}$ is assumed, with α_c at its center. Also, it is assumed that $A(\alpha)$ has full rank and that all the elements $a_{ij}(\alpha)$ have bounded Hessians for all $\alpha \in \Omega_c$.

The function (2.3) being minimized, when $p = 2$, is equivalent to

$$\varphi(\alpha, x) = \frac{1}{2} r^T r + \frac{1}{2}(\alpha - \hat{\alpha})^T D^2 (\alpha - \hat{\alpha}), \tag{3.1}$$

where $r = r(\alpha, x) = b - A(\alpha)x$. The first-order optimality conditions for a local optimum of $\varphi(\alpha, x)$ are the vanishing of the gradients $\nabla_\alpha \varphi$ and $\nabla_x \varphi$, where these gradients are given by

$$\begin{aligned} \nabla_\alpha \varphi &= -J^T(\alpha, x)r + D^2(\alpha - \hat{\alpha}), \\ \nabla_x \varphi &= -A(\alpha)^T r. \end{aligned} \tag{3.2}$$

At each iteration of the SNTLN algorithm with the L_2-norm, we compute the least squares solution to

$$\min_{\Delta x, \Delta \alpha} \left\| M \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} + \begin{pmatrix} -r \\ D(\alpha - \hat{\alpha}) \end{pmatrix} \right\|_2, \tag{3.3}$$

where

$$M = \begin{bmatrix} J(\alpha, x) & A(\alpha) \\ D & 0 \end{bmatrix}. \tag{3.4}$$

The least squares solution is given by

$$M^T M \begin{pmatrix} \Delta\alpha \\ \Delta x \end{pmatrix} = -M^T \begin{pmatrix} -r \\ D(\alpha - \hat{\alpha}) \end{pmatrix} = - \begin{pmatrix} \nabla_\alpha \varphi \\ \nabla_x \varphi \end{pmatrix}, \tag{3.5}$$

where the last equality follows from (3.2). Since M has full rank, by the assumption on $A(\alpha)$, $M^T M$ is nonsingular and $(\Delta\alpha, \Delta x)$ can only be zero when the gradient is zero. Therefore at termination, the algorithm gives (α, x), satisfying the first-order optimality conditions.

Now consider the Hessian of $\varphi(\alpha, x)$ with respect to (α, x). Then,

$$H(\alpha, x) = \begin{pmatrix} \nabla_{\alpha,x}(\nabla_\alpha \varphi(\alpha, x)) & \nabla_{\alpha,x}(\nabla_x \varphi(\alpha, x)) \end{pmatrix} \tag{3.6}$$

$$= \begin{pmatrix} \nabla_{\alpha,x}(-J^T(\alpha, x)r + D^2(\alpha - \hat{\alpha})) & \nabla_{\alpha,x}(-A(\alpha)^T r) \end{pmatrix}. \tag{3.7}$$

Since

$$\nabla_{\alpha,x}(-J^T(\alpha, x)r + D^2(\alpha - \hat{\alpha})) = \begin{pmatrix} J^T(\alpha, x)J(\alpha, x) - \sum_{i=1}^m r_i Q_i(\alpha, x) \\ A^T(\alpha)J(\alpha, x) - \sum_{i=1}^m r_i K_i^T(\alpha) \end{pmatrix} + \begin{pmatrix} D^2 \\ 0 \end{pmatrix} \tag{3.8}$$

and

$$\nabla_{\alpha,x}(-A(\alpha)^T r) = \begin{pmatrix} J^T(\alpha, x) \\ A^T(\alpha) \end{pmatrix} A(\alpha) + \begin{pmatrix} -\sum_{i=1}^m r_i K_i(\alpha) \\ 0 \end{pmatrix}, \tag{3.9}$$

we have

$$H(\alpha, x) = M^T M - \begin{bmatrix} H_1 & H_2 \\ H_2^T & 0 \end{bmatrix},$$

where

$$\begin{aligned} H_1 &= \textstyle\sum_{i=1}^m r_i Q_i(\alpha, x), \\ H_2 &= \textstyle\sum_{i=1}^m r_i K_i(\alpha), \end{aligned}$$

and

$$\begin{aligned} K_i^T(\alpha) &= \text{Jacobian of } i\text{th column of } A^T(\alpha), \\ Q_i(\alpha, x) &= \textstyle\sum_{j=1}^n x_j \nabla_\alpha^2 a_{ij}(\alpha). \end{aligned}$$

Since the Hessian of $a_{ij}(\alpha)$ are all assumed bounded, $\|H_1\|$ and $\|H_2\|$ are of $O(\|r\|)$. Therefore, provided the residual vector r is small, the positive definite matrix $M^T M$ is a good approximation to $H(\alpha, x)$. The iteration (3.5) is therefore a good approximation to Newton's method, and in fact is equivalent to the Gauss–Newton method (see, for example, section 6.1 in [9]). Based on the known convergence properties of the Gauss–Newton method, we can say that the SNTLN algorithm with $p = 2$ will converge to α_c, provided $\|\hat{\alpha} - \alpha_c\|$ and the residual norm $\|r\|$ are both sufficiently small. Convergence of the SNTLN algorithm for $p = 1$ or ∞ is shown elsewhere [26]. To prove convergence, it is necessary to add a line search to the algorithm so that a strict decrease in the norm (2.3) can be shown at each iteration. The algorithm will then converge to a stationary point of (2.3). The computational results obtained so far show, however, that a line search is not required in order to obtain convergence of the algorithm to α_c, provided that the initial parameter estimate $\hat{\alpha}$ is reasonably close to α_c.

We now give explicit expressions for the increments $(\Delta\alpha, \Delta x)$ as computed at each iteration of the SNTLN algorithm. From (3.5) and (3.2) we have

$$\begin{aligned} (J^T J + D^2)\Delta\alpha + J^T A \Delta x &= J^T r - D^2(\alpha - \hat{\alpha}), \\ A^T J \Delta\alpha + A^T A \Delta x &= A^T r. \end{aligned} \tag{3.10}$$

Since $A(\alpha)$ has full rank, we can solve for $\Delta\alpha$ and Δx. These increments are given by

$$(J^T P J + D^2)\Delta\alpha = J^T P r - D^2(\alpha - \hat{\alpha}), \tag{3.11}$$

$$\Delta x = (A^T A)^{-1} A^T (r - J\Delta\alpha), \tag{3.12}$$

where

$$P = I - A(A^T A)^{-1} A^T$$

is the projection onto the orthogonal complement of the range space of A. Note that Δx is given by the least squares solution to the linearized approximation to $r(\alpha + \Delta\alpha, x + \Delta x)$, as given by (2.4). That is

$$\min_{\Delta x} \|r - J\Delta\alpha - A\Delta x\|_2.$$

Also note that the first-order optimality conditions $\nabla_\alpha \varphi = \nabla_x \varphi = 0$ are satisfied when the increments $(\Delta\alpha, \Delta x)$ given by (3.11) and (3.12) are zero. This is seen from (3.2) and the fact that $A^T r = 0$ implies $Pr = r$.

The need to choose positive diagonal elements for the diagonal matrix D, and its relationship to m, n, and s, will now be summarized. Since $\operatorname{rank}(A) = n$, the $m \times m$ projection matrix P has rank $m - n$. Therefore, $\operatorname{rank}(PJ) \leq m - n$. Recall that $\Delta\alpha \in \mathbf{C}^s$, so if $s > \operatorname{rank}(PJ)$ and $D = 0$, the value of $\Delta\alpha$ is not uniquely determined by (3.11). However, since $J^T P J$ is always positive semidefinite, the matrix $J^T P J + D^2$ is always positive definite for any diagonal matrix D with positive diagonal elements, even when the diagonal elements of D are very small. Therefore, (3.11) will always give a unique $\Delta\alpha$ for $D > 0$. If J has full rank $(= s)$ and $s \leq m - n$, then $J^T P J$ is positive definite, and $\Delta\alpha$ is uniquely determined even with $D = 0$. To illustrate the lack of uniqueness in α, when $s > m - n$ and $D = 0$, we give a simple example, where

$m = 3$, $n = 2$, and $s = 2$. Let $r(\alpha, x)$ be given by (2.2) and consider the minimization problem (2.3) with $D = 0$ and $p = 2$. Also, let

$$A(\alpha) = \begin{pmatrix} a_1(\alpha) \\ B \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix},$$

where $a_1(\alpha) = \begin{pmatrix} a_{11} + \alpha_1 & a_{12} + \alpha_2 \end{pmatrix} \in \mathbf{R}^{1\times 2}$, $B \in \mathbf{R}^{2\times 2}$ is a nonsingular, constant matrix, and $\begin{pmatrix} b_2 \\ b_3 \end{pmatrix} \neq 0 \in \mathbf{R}^{2\times 1}$. Choosing $x^* = B^{-1}\begin{pmatrix} b_2 \\ b_3 \end{pmatrix}$ will make $r_2 = r_3 = 0$ and also give

$$r_1 = (b_1 - a_{11}x_1^* - a_{12}x_2^*) - (\alpha_1 x_1^* + \alpha_2 x_2^*).$$

Therefore, any values of α_1 and α_2 that give $r_1 = 0$ will produce a minimum solution to (2.3), with value zero.

The effect of the choice of D on the minimum solution to (2.3) is best understood by the requirement that $\nabla_\alpha \varphi = 0$, with $\nabla_\alpha \varphi$ given by (3.2). To simplify the discussion, let $D = \mu I$, with $\mu > 0$. First we observe that as μ is increased from zero, the term $D(\alpha - \hat{\alpha})$ increasingly dominates the minimization, so the norm $\|\bar{\alpha} - \hat{\alpha}\|$ will go to zero, where $\bar{\alpha}$ is the value of α obtained by the SNTLN algorithm. Therefore, relative large values of μ should be used only when the initial estimate $\hat{\alpha}$ is known to be reliable. The optimality conditions (3.2) require that

$$J^T(\bar{\alpha}, x)r = D^2(\bar{\alpha} - \hat{\alpha}) = \mu^2(\bar{\alpha} - \hat{\alpha}), \tag{3.13}$$

which gives

$$\|r\| \geq \frac{\mu^2}{\|J^T\|}\|\bar{\alpha} - \hat{\alpha}\|. \tag{3.14}$$

It follows from (3.14) that unless a very good initial estimate $\hat{\alpha}$ is known, a value of $\mu \ll \|J^T\|$ should be used.

For problems where $s > m - n$ and a reliable estimate $\hat{\alpha}$ is known, larger values for the elements of D may be chosen. Typical of such problems is STLN, as given by (1.1), where $\hat{\alpha} = 0$. For these cases the elements of D are positive integers giving the multiplicity of the occurrence of each α_i. Other examples include problems where $A(\alpha)$ has a structure (such as Vandermonde) to be preserved, and the values of its elements (possibly subject to error) are known.

4. SNTLN for Vandermonde matrices. In this section, we give a detailed description of the SNTLN algorithm for solving overdetermined systems with a Vandermonde structure,

$$A(\hat{\alpha})x \approx b, \tag{4.1}$$

where

$$A(\alpha) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 \\ \vdots & & & \\ \alpha_1^{m-1} & \alpha_2^{m-1} & \cdots & \alpha_n^{m-1} \end{pmatrix}, \qquad m \geq n, \tag{4.2}$$

and the parameter vector α is

$$\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}.$$

The Vandermonde structure is one of the most frequently occurring nonlinear structures in applications [3, 8, 15, 22]. For example, in the exponential data modeling problems, m uniformly sampled data points y_i are given and are to be fitted to the following model function:

$$y_i \approx \sum_{j=1}^{n} x_j \alpha_j^i = \sum_{j=1}^{n} (a_j e^{\sqrt{-1}\phi_j}) e^{(-d_j + 2\pi\sqrt{-1} f_j) i \Delta t}, \qquad i = 0, \ldots, m-1,$$

where n is the model order and Δt is the constant sampling interval. The objective is to estimate the frequencies f_j, damping factors d_j, amplitudes a_j, and phases ϕ_j, $j = 1, \ldots, n$. The frequencies and damping factors can be found using one of several existing methods, e.g., the linear prediction method with the singular value decomposition [8] or the state–space-based method due to Kung et al. [8, 3, 15], which circumvents polynomial root finding and root selection. Improved versions of both methods, based on total least squares (TLS), are presented in [29]. In [21], we have shown that the STLN method further improves the accuracy of estimated frequencies and damping factors when it is used in the linear prediction method to preserve the Toeplitz structure. Once frequency and damping factors are found by using any of the methods mentioned above, they provide the estimate $\hat{\alpha}$ for the parameter vector α. Then the linear parameters x_j, which contain the amplitudes a_j and phases ϕ_j, $1 \le j \le n$, are estimated from solving the overdetermined Vandermonde system

$$A(\hat{\alpha})x \approx b, \qquad \text{where} \qquad b = \begin{pmatrix} y_0 \\ \vdots \\ y_{m-1} \end{pmatrix}.$$

We may also start the SNTLN iteration without using the methods mentioned above which provide the estimate $\hat{\alpha}$. However, as in any nonlinear problem, a good initial estimate is needed to obtain convergence.

In solving (4.1) using SNTLN, the perturbation on A will be found so that each $\hat{\alpha}_i$ is perturbed to $\alpha_i = \hat{\alpha}_i + h_i$ for some value h_i and the perturbed matrix $A(\alpha)$ keeps the Vandermonde structure. The solution vector x and the perturbation $h = (h_1 \cdots h_n)^T$ on the parameter $\hat{\alpha}$, which satisfy

$$A(\alpha)x = A(\hat{\alpha} + h)x = b - r,$$

will be found while minimizing $\| \begin{pmatrix} r \\ D_h \end{pmatrix} \|_p$.

For the Vandermonde matrix A, the Jacobian $J(\alpha, x)$ for $A(\alpha)x$ is

$$J(\alpha, x) = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ x_1 & x_2 & \cdots & x_n \\ 2\alpha_1 x_1 & 2\alpha_2 x_2 & \cdots & 2\alpha_n x_n \\ \vdots & & & \\ (m-1)\alpha_1^{m-2} x_1 & (m-1)\alpha_2^{m-2} x_2 & \cdots & (m-1)\alpha_n^{m-2} x_n \end{pmatrix}. \tag{4.3}$$

Therefore, in Step 2(a) of Algorithm SNTLN, we need to solve a minimization problem with the $(m+n) \times 2n$ matrix $\begin{pmatrix} A(\alpha) & J \\ 0 & D \end{pmatrix}$, where J and $A(\alpha)$ are given in (4.3) and (4.2), respectively. Note that we have chosen not to perturb the first row of the matrix $A(\alpha)$.

In the next section, we present the numerical test results that compare the performance of SNTLN with $p = 2$ to that of LS and TLS for solving the Vandermonde overdetermined system. We also present results for two problems of the form (2.6) to show the effect of errors in the initial estimate $\hat{\alpha}$ and the ability of SNTLN with $p = 1$ to handle outliers in the data.

5. Computational test results. The SNTLN algorithm has been implemented in MATLAB in order to investigate its computational performance. We denote by SNTLN1 and SNTLN2 the SNTLN algorithm with $p = 1$ and $p = 2$, respectively. First, the accuracy of the solution computed by the SNTLN2 algorithm was compared to that of the LS and TLS methods in solving an overdetermined system $A(\hat{\alpha})x \approx b$, where $A(\hat{\alpha})$ is a Vandermonde matrix. In this test, we assume that the matrix $A(\hat{\alpha})$, i.e., the initial value $\hat{\alpha}$, is given, since this is required in applying the LS and TLS methods. Therefore, the SNTLN2 algorithm is used to solve a linear overdetermined system while preserving the nonlinear structure of the given matrix. Second, we present the computational test results that illustrate the effect on the convergence of the initial choice of $\hat{\alpha}$ and L_p norm on nonlinear parameter estimation problems.

5.1. Comparison of SNTLN2 to LS and TLS for Vandermonde overdetermined system. The computational tests were performed to compare the SNTLN2 solutions with the TLS and LS solutions for Vandermonde overdetermined systems. The initial value for $A(\alpha)$ is assumed to be given, and also it is assumed that there exists a "correct" Vandermonde matrix $A(\alpha_c)$ and vector b_c such that

$$A(\alpha_c)x_c = b_c \tag{5.1}$$

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. Since actual data contains noise, only the perturbed Vandermonde matrix $A(\alpha_c + \delta_\alpha) = A(\hat{\alpha})$ and the perturbed vector b_p are assumed to be known, instead of $A(\alpha_c)$ and b_c. The objective is to compare the three methods LS, TLS, and SNTLN2 in recovering the actual solution x_c by solving the perturbed system $A(\hat{\alpha})x_p \approx b_p$. Specifically, the error vector h for the parameter α, and residual vector r, are computed by solving

$$\min_{r,\alpha} \left\| \begin{pmatrix} r \\ Dh \end{pmatrix} \right\|_2 \qquad \text{such that} \qquad A(\alpha)x_p = A(\hat{\alpha} + h)x_p = b_p - r. \tag{5.2}$$

The test problems are constructed so that $A(\alpha_c)$, b_c, and x_c are known. Then random perturbations δ_α on α_c and δ_b on b_c are generated to give a Vandermonde matrix $A(\alpha_c + \delta_\alpha)$ and $b_p = b_c + \delta_b$, where the components of δ_α and δ_b are uniformly distributed random variables in a given interval. The matrix $A(\alpha)$ and r, x_p, satisfying (5.2) are then computed via LS, TLS, and SNTLN2. For LS, $\alpha = \hat{\alpha}$; i.e., $A(\alpha) = A(\hat{\alpha})$, since the matrix is not perturbed. For TLS, $A(\alpha) = A(\hat{\alpha}) + E$ for some matrix E since TLS does not preserve the structure or take the nonlinear dependence of A on α into account in computing the solution.

In Table 5.1, we present the test results of the following problem. Each data point shown represents the average of 100 solutions, each with different random values in

	(i) b is unperturbed			(ii) | pert. in b_i| $\leq$1.0e-8			(iii) | pert. in b_i| $\leq \gamma$		
γ	LS	TLS	SNTLN2	LS	TLS	SNTLN2	LS	TLS	SNTLN2
1.0e-8	4.8e-8	4.8e-8	4.9e-15	4.5e-8	4.5e-8	2.5e-8	4.6e-8	4.6e-8	2.5e-8
1.0e-6	4.5e-6	4.5e-6	2.2e-16	5.0e-6	5.0e-6	2.5e-8	4.6e-6	4.6e-6	2.5e-6
1.0e-4	4.9e-4	4.9e-4	1.7e-14	5.0e-4	5.0e-4	2.7e-8	4.6e-4	4.6e-4	2.3e-4
1.0e-3	5.0e-3	5.0e-3	3.5e-16	4.2e-3	4.2e-3	2.7e-8	4.7e-3	4.7e-3	2.5e-3
1.0e-2	4.5e-2	4.6e-2	2.1e-14	4.9e-2	4.9e-2	2.4e-8	4.3e-2	4.3e-2	2.7e-2
1.0e-1	4.2e-1	4.2e-1	5.1e-2	4.7e-1	5.7e-1	1.1e-1	5.1e-1	5.1e-1	3.4e-1

TABLE 5.1
Solution error $\frac{\|x_p - x_c\|_2}{\|x_c\|_2}$ *of* x_p *computed by LS, TLS, and SNTLN*2.

the range $[-\gamma, \gamma]$. In the test, $A(\alpha_c)$ is a 15×3 Vandermonde matrix where

$$\alpha_c = \begin{pmatrix} e^{-0.1+2\pi\sqrt{-1}*0.5} \\ e^{-0.2+2\pi\sqrt{-1}*0.4} \\ e^{-0.3+2\pi\sqrt{-1}*0.3} \end{pmatrix}, \qquad x_c = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \qquad b_c = A(\alpha_c)x_c.$$

Then α_c is perturbed by δ_α to give $\hat{\alpha} = \alpha_c + \delta_\alpha$, where the components of δ_α are uniformly distributed random variables in the interval $[-\gamma, \gamma]$. For the perturbation in b_c, we have tested three different cases: (i) when b_c is unperturbed, (ii) b_c is perturbed by uniformly distributed random variables in the interval [-1.0e-8, 1.0e-8], and (iii) b_c is perturbed by uniformly distributed random variables in the interval $[-\gamma, \gamma]$ like α_c. These three cases were tested to study how well the SNTLN2 recovers the correct solution x_c when the given data $\hat{\alpha}$ and b are affected by errors of different kind. The matrix D was chosen to be diag(1.0e-8) in this test.

The iteration in the SNTLN2 algorithm was continued until both

$$\|\Delta\alpha\| \leq \text{e-6 and } \|\Delta x\| \leq \text{e-6}$$

were satisfied. When this convergence test is not satisfied within 20 iterations, then the iteration was terminated and the result obtained at the 20th iteration is taken as the final solution. For most values of γ, this stopping criterion was always satisfied. However, when γ =1.0e-1, in 3 problems for case (i), 8 problems for case (ii), and 7 problems for case (iii), out of 100 test problems each, the iteration was terminated after 20 iterations since the convergence test was not satisfied. This explains the sudden deterioration of the performance of SNTLN2 in tests (i) and (ii) for γ =1.0e-1. However, the performance of SNTLN2 is still better than that of LS or TLS. When the average relative error of SNTLN2 was computed only for the cases converged within 20 iterations, the accuracy was increased to 8.6e-15 and 2.5e-8 in tests (i) and (ii), respectively, even for γ =1.0e-1.

5.2. Effect of errors in data and parameter estimates. As discussed in the previous section, the SNTLN algorithm will converge to the desired minimum vector α_c from an initial estimate $\hat{\alpha}$ if $\hat{\alpha}$ is sufficiently close to α_c. However, the practical question of how large a value of $\|\hat{\alpha} - \alpha_c\|$ will give convergence can only be answered by computational testing. The preliminary computational testing summarized in this section was carried out to explore this property of SNTLN. In addition, we wanted to study two related properties:

1. The effect of data errors on the computed estimate $\overline{\alpha}$ of the parameter vector. Specifically, the effect of the data errors on $\|\overline{\alpha} - \alpha_c\|$.
2. The robustness of the L_1-norm with respect to outliers in the data.

ϵ	Outlier Norm	$\|\overline{\alpha} - \alpha_c\|/\|\alpha_c\|$ from SNTLN2	$\|\overline{\alpha} - \alpha_c\|/\|\alpha_c\|$ from SNTLN1
0	5e-3	9.6e-3	0
5e-9	5e-3	9.6e-3	2.7e-7
5e-8	5e-3	9.6e-3	2.7e-6
5e-7	5e-3	9.6e-3	1.7e-5
5e-6	5e-3	9.6e-3	1.6e-4
5e-5	5e-3	1.1e-2	2.1e-3
5e-5	0	9.0e-3	1.9e-3

TABLE 5.2
Effect of errors in data on parameter estimate—type 1 *Signal.*

We now present computational results which show the ability of the SNTLN algorithm to determine good values of the parameter vector α and the coefficient vector x, in spite of noise in the data and relatively poor estimates of α_c. In order to carry out these computational tests, two different parameter estimation problems were used. For each test problem it is assumed that a noiseless signal $f(t)$ is of the form given below. The measured signal at m values of t is assumed to have the form

$$f_i = f(t_i) + \eta_i, \qquad i = 1, \ldots, m, \tag{5.3}$$

where the η_i represent noise or error in the measurement. The following two types of signal were chosen:

1. $f(t) = \sum_{j=1}^{n} x_j e^{-\alpha_j t}$,
2. $f(t) = \sum_{j=1}^{n} x_j e^{-(t-\alpha_j)^2/\sigma^2}$.

Given the corresponding noisy data $f_i, i = 1, \ldots, m$, it is desirable to get the best estimate of the true parameter vector α_c and linear coefficient vector x_c, which determine the undistorted signal $f(t)$. We also know an initial estimate $\hat{\alpha}$ of α_c, which may also be in error.

The data for the signal of type 1 was obtained from that used by Osborne and Smyth [19]. Specifically, the values $\alpha_c = (\,0 \quad 4 \quad 7\,)^T$ and $x_c = (\,0.5 \quad 2 \quad -1.5\,)^T$ were used to give $f(t)$ over the interval $t \in [0, 1]$. A total of 30 points t_i were used so that $m = 30$, $n = 3$, and $s = 3$.

The sum of Gaussian functions, type 2 signal, is similar to that used in [13]. For this test, the values $\sigma^2 = 0.05$, $\alpha_c = (0.1\ 0.3\ 0.5\ 0.9)^T$, and $x_c = (1.0\ 0.5\ 2.0\ 0.25)^T$ were used. The values of α_j and x_j were to be determined (σ is assumed to be known). The 64 values of t_i were chosen to be equally spaced in $[0, 1]$ so that $m = 64$, $n = 4$, and $s = 4$. Since $n + s < m$ for both signal types, a small value (1.0e-8) was again chosen for the diagonal elements of D.

We used the type 1 signal to measure the effect of errors in the data vector f_i on the computed parameter estimate $\bar{\alpha}$. First, uniformly distributed random errors η_i, $i = 1, \ldots, m$, in the interval $[-\epsilon, \epsilon]$ were added to $f(t_i)$ to give f_i as in (5.3). With ϵ =5.0e-5, the relative error $\frac{\|\bar{\alpha}-\alpha_c\|}{\|\alpha_c\|}$ was 1.9e-3 using SNTLN1 and 9.0e-3 using SNTLN2. This is shown in the last row of Table 5.2. Additional tests with uniformly distributed random errors showed that the relative error in $\bar{\alpha}$ is proportional to ϵ for both $p = 1$ and $p = 2$ and that the relative error for $p = 1$ is somewhat less than the relative error for $p = 2$. For $\epsilon = 0$, as expected, we get $\bar{\alpha} = \alpha_c$. The initial estimate $\hat{\alpha} = \alpha_c$ was used for all these tests, so the error in $\bar{\alpha}$ was due entirely to the error in the data.

The next set of tests were to determine the effect of outliers in the data on the computed parameter estimate. In addition to the random errors η_i at each t_i, a single outlier was introduced at one of the points t_i. Its magnitude was 5.0e-3. The parameter estimate $\bar{\alpha}$ was then computed for a sequence of increasing values of ϵ. The value of ϵ was increased from 5.0e-9 to 5.0e-5. For each value of ϵ, the parameter estimate was computed using SNTLN with $p = 1$ and $p = 2$. The results are summarized in the first six rows of Table 5.2. It is seen that for $p = 2$, the effect of the outlier dominated the error in the parameter estimate. In contrast, the use of SNTLN1 produces a parameter estimate which ignores the outlier and depends only on the size of the random errors η_i. In particular, the parameter estimate error is essentially proportional to ϵ and is the same whether an outlier is present or not. Changing the location of the outlier had only a small effect on these results. Changing the magnitude of the outlier had no effect on the results with $p = 1$. This shows that SNTLN1 is very robust with respect to a single large error in the data.

Additional tests were made using more than one outlier with the type 1 signal. Outliers were introduced at randomly chosen time points t_i, with all other values of $f_i = f(t_i)$, that is, without error. Surprisingly, it was found that up to 10 outliers could be added, in some cases, with no adverse effect on the estimate $\overline{\alpha}$, using $p = 1$. That is, SNTLN1 almost always gave $\overline{\alpha} = \alpha_c$. However, when there were more than 10 outliers, the error $\|\overline{\alpha} - \alpha_c\|$ was comparable to the norm of the outliers.

The effect of error in the initial parameter estimate $\hat{\alpha}$ was investigated using a signal of type 2. In addition, the robustness, with respect to outliers, of SNTLN1 was further confirmed with this larger, and very different, type of test problem.

The ability of both SNTLN1 and SNTLN2 to converge to α_c from different initial parameter estimates $\hat{\alpha}$ was tested. This was done by choosing

$$\hat{\alpha} = \alpha_c + \delta, \tag{5.4}$$

where each δ_i, $i = 1, \ldots q$, is a uniformly distributed random variable in the interval $[-\gamma, \gamma]$. In addition, a uniformly distributed random error η_i was added to f_i, with $|\eta_i| \leq$e-7, for all cases. Convergence to α_c becomes more difficult as γ increases. For the purposes of this test we say SNTLN has converged if both $\|\Delta\alpha\| \leq$e-6 and $\|\Delta x\| \leq$e-6, in 10 iterations or less. The results for SNTLN1 and SNTLN2 are shown in Figure 5.1. Each data point shown represents the result of 20 solutions, each with different random values of the δ_i. The percentage of the solutions which converged is plotted as a function of the maximum value of $|\delta_i|$, as given by γ. Six values of γ were used, $\gamma = (0, .01, .02, .03, .05, .07)$, and since the smallest $\alpha_1 = 0.1$, the initial estimate error in α_1 could be as large as 70%. The upper plot (denoted by o) shows the results for no outliers and represents both SNTLN1 and SNTLN2. It is seen that the percentage of convergence is the same for both, and in fact, all cases converged for $\gamma \leq .02$. For $\gamma = .07$, the percentage of convergence drops to 75%. For no outliers, the error in the final computed estimate $\bar{\alpha}$, is determined by the random errors η_i in f_i and was never greater than 3.3e-6 for either SNTLN1 or SNTLN2.

In order to investigate the effect of outliers, the values of γ used above were repeated, with 10 and then 25 outliers. The outliers each had a value of ± 0.1 and were added to f_i (in addition to η_i) at randomly selected positions. Thus the total error in the signal f_i at t_i consisted of a small random perturbation η_i at every point t_i and a much larger outlier at either 10 or 25 of the 64 total points. The results are shown in the lower two curves in Figure 5.1. The SNTLN2 algorithm did not satisfy the convergence criterion in many cases so the results shown represent the SNTLN1 algorithm only.

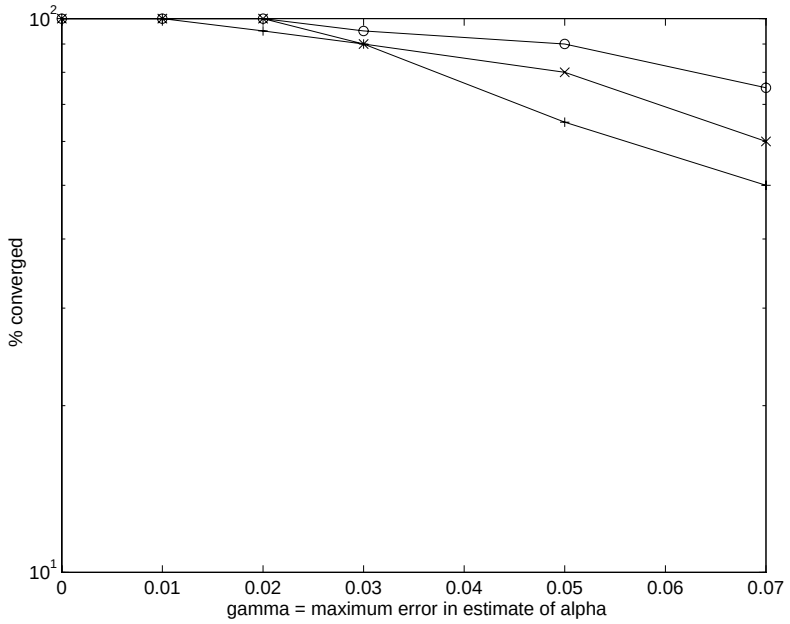

FIG. 5.1. *Effect of initial $\hat{\alpha}$ estimate error and number of outliers on convergence of SNTLN*1 *and SNTLN*2. *Random error in f = 1.0e-7, outlier magnitude = 0.1. Key:* o *(no outliers, L*1 *and L*2*),* × *(*10 *outliers, L_1 only),* + *(*25 *outliers, L_1 only). Maximum number of iterations for convergence: L_1 (*6 *iterations) and L_2 (*9 *iterations). Maximum error in computed α =* 7.8e-6*, using SNTLN*1.

The results show that for 10 outliers, the percentage of convergence is 100% for $\gamma \leq .02$ and drops to 60% for $\gamma = .07$. For 25 outliers, the percentage of convergence is 100% for $\gamma \leq .01$, and it drops to 50% for $\gamma = .07$. It is important to note that the converged value of $\bar{\alpha}$ never had an error greater than 7.8e-6, even with 25 outliers. This means that when SNTLN1 converges, the error in the final estimate of α is determined completely by the values of the random errors η_i and is *unaffected by the outliers*. Furthermore, when convergence took place it never required more than 6 iterations of SNTLN1. Based on these computational results, it appears that SNTLN1 is very robust with respect to outliers, provided that the total number of outliers is somewhat less than $(m-n)/2$. The explanation of this robust behavior is being investigated, both theoretically and computationally [26].

Acknowledgments. We would like to thank Mr. Lei Zhang for carrying out the numerical tests, and the anonymous reviewers for valuable comments which made it possible to improve the paper.

REFERENCES

[1] T. J. ABATZOGLOU, J. M. MENDEL, AND G. A. HARADA, *The constrained total least squares technique and its application to harmonic superresolution*, IEEE Trans. Signal Processing, 39 (1991), pp. 1070–1087.

[2] A. A. ANDA AND H. PARK, *Self-scaling fast rotations for stiff least squares problems*, Linear Algebra Appl., 234 (1996), pp. 137–161.

[3] H. BARKHUIJSEN, R. DE BEER, AND D. VAN ORMONDT, *Improved algorithm for noniterative time-domain model fitting to exponentially damped magnetic resonance signals*, J. Mag-

netic Resonance, 73 (1987), pp. 553–557.
[4] I. Barrodale and A. Young, *Algorithms for best L_1 and L_∞ linear approximations on a discrete set*, Numer. Math., 8 (1966), pp. 295–306.
[5] I. Barrodale, *L_1 approximation and the analysis of data*, Appl. Statistics, 17 (1968), pp. 51–57.
[6] D. Bates and M. Lindstrom, *Nonlinear least squares with conditionally linear parameters*, in Proceedings of the Statistical Computing Section, American Statistical Association, Washington, DC, 1986, pp. 152–157.
[7] H. Chen, S. Van Huffel, and J. Vandewalle, *Exponential Data Fitting Using the Structured Total Least Norm Technique*, Technical report TR 95-18, Departement Elektrotechniek ESAT-SISTA, University of Leuven, Belgium, March, 1995.
[8] R. De Beer and D. Van Ormondt, *Analysis of NMR data using time-domain fitting procedures*, in In-vivo Magnetic Resonance Spectroscopy I: Probeheads, Radiofrequency Pulses, Spectrum Analysis, NMR Basic Principles and Progress 26, M. Rudin, ed., Springer-Verlag, Berlin, Heidelberg, 1992, pp. 201–248.
[9] R. Fletcher, *Practical Methods of Optimization*, John Wiley, New York, 1987.
[10] G. H. Golub and V. Pereyra, *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate*, SIAM J. Numer. Anal., 10 (1973), pp. 413–432.
[11] T. A. Grandine, *Generating Surface Lofts to Scattered Data*, Engineering Computing and Analysis Technical report ECA-TR-157, Boeing Computer Services, Seattle, WA, 1991.
[12] L. Kaufman, *A variable projection method for solving separable nonlinear least squares problems*, BIT, 15 (1975), pp. 49–57.
[13] L. Kaufman and G. Sylvester, *Separable nonlinear least squares with multiple right-hand sides*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 68–89.
[14] R. Kumaresan and D. W. Tufts, *Estimating the parameters of exponentially damped sinusoids and pole-zero modeling in noise*, IEEE Trans. Acoust. Speech Signal Proc., 30 (1982), pp. 833–840.
[15] S. Y. Kung, K. S. Arun, and D. V. Bhaskar Rao, *State-space and singular value decomposition-based approximation methods for the harmonic retrieval problem*, J. Opt. Soc. Amer., 73 (1983), pp. 1799–1811.
[16] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice–Hall, Englewood Cliffs, NJ, 1974.
[17] Yuying Li, *Solving L_p-norm Problems and Applications*, Cornell Theory Center report CTC93TR122, Cornell University, Ithaca, NY, 1993.
[18] M. R. Osborne, *Some special nonlinear least squares problems*, SIAM J. Numer. Anal., 12 (1975), pp. 571–592.
[19] M. R. Osborne and G. K. Smyth, *A modified Prony algorithm for exponential function fitting*, SIAM J. Sci. Comput., 16 (1995), pp. 119–138.
[20] H. Park, J. B. Rosen, and J. Glick, *Structure preserving total least norm method and application to parameter estimation*, in Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing, Vol. 2, 1995, pp. 1141–1144.
[21] H. Park, J. B. Rosen, and S. Van Huffel, *Structure preserving total least squares method and its application to parameter estimation*, SVD and Signal Processing, III: Algorithms, Architectures and Applications, M. Moonen and B. De Moor, eds., Elsevier, New York, 1995, pp. 399–406.
[22] M. A. Rahman and K. B. Yu, *Total least squares approach for frequency estimation using linear prediction*, IEEE Trans. Acous. Speech Signal Proc., 35 (1987), pp. 1440–1454.
[23] J. R. Rice and J. S. White, *Norms for smoothing and estimation*, SIAM Rev., 6 (1964), pp. 243–256.
[24] 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. 110–128.
[25] J. B. Rosen, H. Park, and J. Glick, *Total least norm for linear and nonlinear structured problems*, in Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling, S. Van Huffel, ed., SIAM, Philadelphia, 1996, pp. 203–214.
[26] J. B. Rosen, H. Park, J. Glick, and L. Zhang, *Accurate Solution to Overdetermined Systems with Errors, Using L_1 Norm Minimization*, Tech. report 98-7, Department UCSD, La Jolla, CA, January 13, 1988 (submitted for publication).
[27] H. Späth and G. A. Watson, *On orthogonal linear l_1 approximation*, Numer. Math., 51 (1996), pp. 531–543.
[28] S. Van Huffel, H. Park, and J. B. Rosen, *Formulation and solution of structured total least norm problems for parameter estimation*, IEEE Trans. Signal Process., (1996), pp. 2464–2474.

[29] S. Van Huffel, L. Aerts, J. Bervoets, J. Vandewalle, C. Decanniere, and P. Van Hecke, *Improved quantitative time-domain analysis of NMR data by total least squares*, in Signal Processing *VI*: Theories and applications, J. Vandewalle, R. Boite, M. Moonen and A. Oosterlinck, eds., Elsevier–North Holland, 1992, pp. III:1721–1724.

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

[31] G. A. Watson and K. F. C. Yiu, *On the solution of the errors in variables problem using the* l_1 *norm*, BIT, 31 (1991), pp. 697–710.