Journal of Computational Mathematics, Vol.21, No.4, 2003, 463–472.

A SUCCESSIVE LEAST SQUARES METHOD FOR STRUCTURED TOTAL LEAST SQUARES [*1)]

Plamen Y. Yalamov
(*Center of Applied Mathematics and Informatics, University of Rousse, 7017 Rousse, Bulgaria*)

Jin-yun Yuan
(*Departamento de Matemática, Universidade Federal do Paraná, Centro Politécnico, Caixa Postal 19.081, 81531-990 Curitiba – PR, Brazil*)

Abstract

A new method for Total Least Squares (TLS) problems is presented. It differs from previous approaches and is based on the solution of successive Least Squares problems. The method is quite suitable for Structured TLS (STLS) problems. We study mostly the case of Toeplitz matrices in this paper. The numerical tests illustrate that the method converges to the solution fast for Toeplitz STLS problems. Since the method is designed for general TLS problems, other structured problems can be treated similarly.

Key words: Structure total least squares, Linear Least squares, Successive linear squares method, Toeplitz systems, Structure least squares.

1. Introduction

Total Least Squares (TLS) problems appear in many engineering applications such as signal and image processing, systems identification, and systems response prediction. A good survey of areas of application and computational methods is given in [17].

The TLS problem can be stated as follows:

$$\|E \mid r\|_F = \min, \text{ where } (A+E)x = b+r. \tag{1}$$

Here $A, E \in \mathcal{R}^{m\times n}$ (usually $m \geq n$), and $x \in \mathcal{R}^n, b, r \in \mathcal{R}^m$. The subscript F denotes the Frobenius norm. E and r are called errors in the model.

All the algorithms in [17] are based on the Singular Value Decomposition (SVD) analysis (see [9, 10]). Other approaches are taken in [3] (general matrices) and [12] (Toeplitz matrices) where methods for nonlinear equations are used to solve the problem. All these methods are suitable for general matrices, and do not take into account any structure in the matrix E. Very often in practice the matrix A has some structure, e. g., Toeplitz, or Hankel [13]. Sometimes the matrix E requires to have the same structure as A. We will call this problem a Structured TLS (STLS) problem. For this problem the SVD based methods of [17], and the methods of [3, 12] do not produce a matrix E with the desired structure.

A different approach is applied in [15] where minimization techniques are used to solve STLS problems. Toeplitz and sparsity structures are considered as an application. This method produces a matrix E with a prescribed structure. The method is suitable only for TLS problems of small size.

[*] Received March 13, 2001; final revised January 15, 2002.

[1)] This research was started when the first author visited the Federal University of Paraná, Curitiba, Brazil and finished when the second author visited Stanford University, USA. The work of the first author was also supported by Grant MM-707/97 from the National Scientific Research Fund of the Bulgarian Ministry of Education and Science. The work of the second author was partially supported by CNPq, CAPES, FINEP, Fundação Araucaria and FUNPAR, Brazil.

A number of methods for solving large Structured LS (SLS) problems ([2, 4, 5, 6, 7]) motivates us to establish some method for solving STLS problems by using SLS methods. The purpose of this paper is to propose such a method for the STLS problem in which the basic kernel is the solution of a LS problem. In this way the proposed method can be used for solving large STLS problems. We give a general framework of the method. Then, by a suitable choice of a parameter, the method is applicable to structured, or unstructured problems (We use the same idea as in [15]). We prove global convergence for any structure. In the case of Toeplitz A and E we show also that each iteration step is faster than one step of the method in [15]. While the minimization of the errors in [15] is with respect to the 1, 2, and infinity norms, here we discuss only the 2-norm. Clearly, this norm is the best choice when LS solutions are involved. In this paper, the existence of solution of the structured total least squares problem (1) always assumed. The outline of the paper is as follows. In Section 2 we present the new method and study its convergence. In Section 3 the implementation for Toeplitz STLS problems is considered. Finally, numerical experiments are give in Section 4.

2. The LS Method

Since the equation

$$(A+E)x = b + r \tag{2}$$

is nonlinear with respect to the unknowns E, x, and r, we assume that the unknowns in a nonlinear system can be split into two groups, for example, one group for x and another group for E and r. With this splitting, if the unknowns in one of the groups are constants the problem becomes linear with respect to the unknowns from the other group, and vice versa.

In such nonlinear problems we can start with some initial value for one of the groups of unknowns, and then alternatively compute approximations of the two groups of unknowns by solving linear problems according to some iteration scheme.

For the TLS problem we suggest the LS solution $x^{(0)}, Ax^{(0)} = b + r^{(0)}$, as an initial value for x. The same initial value is chosen also in [3, 15]. This choice is natural because the LS problem is just a special case of the TLS problem, and in many cases $x^{(0)}$ will be close enough to the solution of the TLS problem.

Let us also note that if x is constant, and E and r are variables then problem (1) can be rewritten as

$$\left\| \begin{matrix} r \\ \alpha \end{matrix} \right\|_2 = \min, \quad Ax + X\alpha = b + r. \tag{3}$$

Here the matrix X and vector α are chosen in such a way that

$$X\alpha = Ex.$$

This choice depends on the structure of the matrix E. We present a few examples:

- E is unstructured. Then we have

$$X = \begin{pmatrix} x_1 & \cdots & x_n & & & & & & \\ & & & x_1 & \cdots & x_n & & & \\ & & & & & & \ddots & & \\ & & & & & & x_1 & \cdots & x_n \end{pmatrix} \in \mathcal{R}^{m \times mn},$$

$$\alpha =$$

$$\mathrm{Vec}(E) = (e_{11}, e_{12}, \ldots, e_{1n}, e_{21}, e_{22}, \ldots, e_{2n}, \ldots, e_{m1}, e_{m2}, \ldots, e_{mn})^T.$$

- E is general sparse. Then X and α are also sparse, and their sparsity pattern depends on the sparsity pattern of E.

- E is general Toeplitz. Then we have

$$X = \begin{pmatrix} x_n & x_{n-1} & \cdots & x_1 & & & 0 \\ & x_n & \cdots & x_2 & x_1 & & \\ & & \ddots & & \ddots & \ddots & \\ 0 & & & x_n & \cdots & x_2 & x_1 \end{pmatrix} \in \mathcal{R}^{m\times(m+n-1)},$$

$$\alpha = (e_{n-1} \ldots e_1 \; e_0 \; e_{-1} \ldots e_{-m+1})^T \in \mathcal{R}^{m+n-1}.$$

- E is banded Toeplitz. In this case we get

$$X = \begin{pmatrix} x_{u+1} & x_u & \cdots & x_1 & & & \\ x_{u+2} & x_{u+1} & \cdots & x_2 & x_1 & & \\ \vdots & & & & \ddots & \ddots & \\ \vdots & & & & & & x_1 \\ x_n & & & & & & x_2 \\ & \ddots & & & & & \vdots \\ & & x_n & & & & x_{n-l+2} \end{pmatrix} \in \mathcal{R}^{m\times(u+l+1)},$$

$$\alpha = (e_u, \ldots, e_1, e_0, e_{-1}, \ldots, e_{-l})^T,$$

where u is the upper bandwidth (number of nonzero super-diagonals), and l is the lower bandwidth (number of nonzero sub-diagonals) of matrix E.

Then (3) is clearly equivalent to

$$\left\| \begin{matrix} r \\ \alpha \end{matrix} \right\|_2 = \min,$$

$$(-I \; X)\begin{pmatrix} r \\ \alpha \end{pmatrix} = b - Ax, \tag{4}$$

i. e. the couple (r, α) (or the entries of (E, r), respectively) is the minimum-norm solution of the underdetermined linear system (4). Here $I \in \mathcal{R}^{m\times m}$ denotes the identity matrix, and $X \in \mathcal{R}^{m\times k}$, where k depends on the structure of the matrix E. We can decompose every vector

$$\begin{pmatrix} r \\ \alpha \end{pmatrix} = \begin{pmatrix} -I \\ X^T \end{pmatrix} y + z$$

where z is belong to the null space of $(-I,\, X)$. Then, the solution of (4) is $\begin{pmatrix} r \\ \alpha \end{pmatrix} = (-I,\, X)^T y$ where y satisfies (5) (see [2]).

Let us denote $s = b - Ax$ for brevity. Let (r, α) be the LS solution of problem (4). One of the ways to find this solution is to solve

$$(I + XX^T)y = s, \tag{5}$$

and then set

$$r = -y, \quad \alpha = X^T y.$$

We discuss this method of solution because the matrix $I + XX^T$ is not difficult to compute in practical applications, and has some nice properties. Let us note that $I + XX^T$ is symmetric positive definite (s.p.d.), and its smallest eigenvalue is not less than 1. So, when it is necessary to solve system (5) we can use conjugate gradients with a suitable preconditioner.

Now we give the matrix $I + XX^T$ and the vector α for the four example structures above:

- E is unstructured:

$$I + XX^T = (1 + \|x\|_2^2)I, \quad \alpha_{(i-1)n+j} = y_i x_j \quad \text{(in this case } E = yx^T\text{)}.$$

- E is general sparse:

$$I + XX^T = \mathrm{diag}\{(1 + \sum_{p=1}^{q(1)} x_{j_p}^2), \dots, (1 + \sum_{p=1}^{q(m)} x_{j_p}^2)\},$$
$$\alpha_{(i-1)n+j_p} = y_i x_{j_p},$$

  where $j_1, j_2, \dots, j_{q(i)}$ are the positions of the nonzero entries in row i of matrix E, and $q(i)$ is the number of nonzero entries of E in row i.

- E is general Toeplitz:

$$I + XX^T =$$
$$\mathrm{toeplitz}([1 + \sum_{i=1}^{n} (x_i^{(k-1)})^2], \quad \sum_{i=1}^{n-1} x_i^{(k-1)} x_{i+1}^{(k-1)},$$
$$\sum_{i=1}^{n-2} x_i^{(k-1)} x_{i+2}^{(k-1)}, \quad \dots, x_1^{(k-1)} x_n^{(k-1)}, \quad 0, \dots, 0),$$
$$\alpha = X^T y,$$

  where toeplitz(c) denotes a symmetric Toeplitz matrix whose first row is c, and α is a product of a Toeplitz matrix and a vector. The matrix $I + XX^T$ is also s.p.d. and banded (with $2n+1$ nonzero diagonals).

- E is banded Toeplitz. In this case matrix $I + XX^T$ is banded with lower bandwidth $2l$, and upper bandwidth $2u$; α is the product of this banded matrix and a vector.

Let us note that in the last example matrix $I + XX^T$ is not Toeplitz but is close to Toeplitz. The difference is in the first $\max(u, l)$ rows. The rest of the rows form a Toeplitz submatrix.

So, when x is given we can compute E and r. When E is given we can compute x and r by solving a LS problem with the matrix $A + E$, and the right hand side b. Based on these notes we propose the following TLS method:

Algorithm 1.

```
Solve the LS problem $Ax = b$ to get $x^{(0)}$, and $r^{(0)} = -s^{(0)} = Ax^{(0)} - b$
for $k = 1, 2, \dots$ until convergence
    Solve the LS problem $(-I \ X^{(k-1)}) \begin{pmatrix} y^{(k)} \\ \alpha^{(k)} \end{pmatrix} = s^{(k-1)}$    (*)
    Define $E^{(k)}$ such that $X^{(k-1)}\alpha^{(k)} = E^{(k)}x^{(k-1)}$
    Solve the LS problem $(A + E^{(k)})x = b$ to get $x^{(k)}$    (**)
    $s^{(k)} = b - Ax^{(k)}$
    $r^{(k)} = E^{(k)}x^{(k)} - s^{(k)}$
end
```

The first important issue is whether we compute satisfactory E and r. The following theorem shows that the iterations reduce $\|(r^T \ \alpha^T)^T)\|_2$ at each step.

Theorem 1 *If $\alpha^{(k)}, r^{(k)}$ and $\alpha^{(k-1)}, r^{(k-1)}$ are the errors from two successive iterations, we have*

$$\left\| \begin{matrix} r^{(k)} \\ \alpha^{(k)} \end{matrix} \right\|_2 \leq \left\| \begin{matrix} r^{(k-1)} \\ \alpha^{(k-1)} \end{matrix} \right\|_2 .$$

Equality holds if and only if $\alpha^{(k)} = \alpha^{(k-1)}$ and $r^{(k)} = r^{(k-1)}$.

Proof. From the way we compute $E^{(k)}$ and Algorithm 1 we have

$$\begin{aligned} (A+E^{(k)})x^{(k-1)} &= Ax^{(k-1)} + E^{(k)}x^{(k-1)} \\ &= b - s^{(k-1)} + X^{(k-1)}\alpha^{(k)} \\ &= b - s^{(k-1)} + X^{(k-1)}(X^{(k-1)})^T y^{(k)} \\ &= b - y^{(k)}. \end{aligned} \tag{6}$$

Then we have

$$\left\| \begin{array}{c} r^{(k)} \\ \alpha^{(k)} \end{array} \right\|_2^2 = \|r^{(k)}\|_2^2 + \|\alpha^{(k)}\|_2^2 \leq \|y^{(k)}\|_2^2 + \|(X^{(k-1)})^T y^{(k)}\|_2^2, \tag{7}$$

as far as $r^{(k)}$ and $y^{(k)}$ are residuals for the system $(A+E^{(k)})x = b$ but $r^{(k)}$ is the residual from the LS solution, and $\alpha^{(k)} = (X^{(k-1)})^T y^{(k)}$. From (7) we get

$$\begin{aligned} \left\| \begin{array}{c} r^{(k)} \\ \alpha^{(k)} \end{array} \right\|_2^2 &\leq (y^{(k)})^T y^{(k)} + (y^{(k)})^T X^{(k-1)}(X^{(k-1)})^T y^{(k)} \\ &= (y^{(k)})^T \left[I + X^{(k-1)}(X^{(k-1)})^T \right] y^{(k)} \\ &= (s^{(k-1)})^T \left[I + X^{(k-1)}(X^{(k-1)})^T \right]^{-1} s^{(k-1)} \\ &= (X^{(k-1)}\alpha^{(k-1)} - r^{(k-1)})^T \left[I + X^{(k-1)}(X^{(k-1)})^T \right]^{-1} \\ &\quad (X^{(k-1)}\alpha^{(k-1)} - r^{(k-1)}) \\ &= [(r^{(k-1)})^T \ (\alpha^{(k-1)})^T] Z \left(\begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right), \end{aligned} \tag{8}$$

where

$$Z = \left(\begin{array}{c} -I \\ (X^{(k-1)})^T \end{array} \right) \left[I + X^{(k-1)}(X^{(k-1)})^T \right]^{-1} (-I \ X^{(k-1)}).$$

Let us assume that the SVD of $(-I \ X^{(k-1)})$ is given as

$$(-I \ X^{(k-1)}) = U\Sigma V^T,$$

where $U \in \mathcal{R}^{m\times m}, \Sigma \in \mathcal{R}^{m\times(2m+n-1)}, V \in \mathcal{R}^{(2m+n-1)\times(2m+n-1)}$, and U and V are orthogonal. Then after some standard manipulation we get

$$Z = V_m V_m^T, \tag{9}$$

where V_m consists of the first m columns of V. Hence, from (8) and (9) we obtain

$$\left\| \begin{array}{c} r^{(k)} \\ \alpha^{(k)} \end{array} \right\|_2^2 \leq \left\| V_m^T \left(\begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right) \right\|_2^2 \leq \left\| \begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right\|_2^2. \tag{10}$$

For the second part of the proof we will prove only that

$$\left(\begin{array}{c} r^{(k)} \\ \alpha^{(k)} \end{array} \right) = \left(\begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right), \text{ if } \left\| \begin{array}{c} r^{(k)} \\ \alpha^{(k)} \end{array} \right\|_2 = \left\| \begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right\|_2. \tag{11}$$

The statement in the opposite direction is evident.

From (11) it follows that everywhere in (7) and (10) we have equalities instead of inequalities. First, from the equality in (7) we have that $r^{(k)} = -y^{(k)}$ because the LS problem has a unique solution and $\|r^{(k)}\|_2 = \|y^{(k)}\|_2$. Second, from the equality in (10) we have that

$$\left(\begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right)$$

is the right singular vector of V_m^T corresponding to its largest singular value which is clearly 1 (because V_m^T is a block from an orthogonal matrix). Evidently, the SVD of V_m^T is

$$V_m^T = (I \ 0)V^T,$$

so, its first right singular vector is $v^{(1)}$, the first column of V. Thus,

$$\left(\begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right) = cv^{(1)}, \quad c = \left\| \begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right\|_2 . \tag{12}$$

Now let us note that

$$\left(\begin{array}{c} -y^{(k)} \\ \alpha^{(k)} \end{array} \right)$$

is a minimum norm solution of the underdetermined system in Algorithm 4. Then we get

$$\left(\begin{array}{c} -y^{(k)} \\ \alpha^{(k)} \end{array} \right) \quad = \quad \left(\begin{array}{c} r^{(k)} \\ \alpha^{(k)} \end{array} \right) = \left(\begin{array}{c} -I \\ (X^{(k-1)})^T \end{array} \right) (I + X^{(k-1)}(X^{(k-1)})^T)^{-1}$$

$$(-I \; X^{(k-1)}) \left(\begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right) = Z \left(\begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right) . \tag{13}$$

Here we used the expression for

$$s^{(k-1)} = X^{(k-1)}\alpha^{(k-1)} - r^{(k-1)} = E^{(k-1)}x^{(k-1)} - r^{(k-1)}.$$

Then from (9), (12) and (13) we get

$$\left(\begin{array}{c} r^{(k)} \\ \alpha^{(k)} \end{array} \right) cV_m V_m^T v^{(1)} = cv^{(1)} = \left(\begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array} \right) . \quad \diamond$$

Remark. Let us note that when E and r are not structured then

$$\left\| \begin{array}{c} r \\ \alpha \end{array} \right\|_2 = \|E \mid r\|_F ,$$

and we minimize the errors in the classical sense. Also note that the algorithm will give the least squares solution for no TLS solution problem (1).

This theorem shows that the sequence $\{(r^{(k)}, \alpha^{(k)})\}$ converges and the iteration reduces the norm of the error at each step until the errors of two successive steps become very close. This suggests stopping criteria of the form

$$\left\| (\begin{array}{c} r^{(k)} - r^{(k-1)} \\ \alpha^{(k)} - \alpha^{(k-1)} \end{array}) \right\|_2 < \varepsilon ,$$

or,

$$\left\| (\begin{array}{c} r^{(k-1)} \\ \alpha^{(k-1)} \end{array}) \right\|_2 - \left\| (\begin{array}{c} r^{(k)} \\ \alpha^{(k)} \end{array}) \right\|_2 < \varepsilon , \tag{14}$$

where ε is a given tolerance.

Now let us discuss in more detail the expression $\|E^{(k)}|r^{(k)}\|_F$ for general matrices A which we minimize. At the point of minimum $(E \mid r)$ we have

$$\|E \mid r\|_F^2 = \|yx^T| - y\|_F^2 = \|y\|_2^2(1 + \|x\|_2^2) = \|s\|_2^2/(1 + \|x\|_2^2), \tag{15}$$

where $(A + E)x = b + r$. But the right hand side in (15) is exactly the Rayleigh Quotient (RQ)

$$\rho = \frac{s^T s}{1 + x^T x} = \frac{(b^T - x^T A^T)(b - Ax)}{1 + x^T x},$$

which is well-known to be the solution of the eigenvalue problem

$$\left(\begin{array}{cc} A^T A & b^T b \\ b^T A & b^T b \end{array} \right) \left(\begin{array}{c} x \\ -1 \end{array} \right) = \lambda \left(\begin{array}{c} x \\ -1 \end{array} \right) . \tag{16}$$

But the solution x of (16) is the TLS solution when $\lambda = \sigma_{n+1}^2$ (for example, see [17, p. 37]), and σ_{n+1} is the smallest singular value of $(A \mid b)$. So, Algorithm 1 is similar to the RQ iteration for the solution of the TLS problem proposed in [3] in that both start the iteration with the LS solution and try to minimize a RQ. Thus the same remarks about the convergence as in [3] are valid. More precisely, we cannot say to which eigenvalue of (16) Algorithm 1 will converge.

However, it can be shown (see also [16]) that starting from the LS solution $x^{(0)}$ convergence to σ^2_{n+1} is guaranteed if

$$\mu_0 \le \frac{1}{4}(\sigma^2_n - \sigma^2_{n+1}),$$

where

$$\mu_0 = \frac{\|r^{(0)}\|_2}{(1+\|x^{(0)}\|_2^2)^{1/2}},$$

and σ_n is the smallest singular value of $(A \mid b)$ greater than or equal to σ_{n+1}.

Unfortunately, for structured matrices we get a RQ which is a solution to a nonlinear problem of the type (16). The solution of this problem is much more complicated than Algorithm 1 proposed in this paper, and we will not discuss it.

We did some experiments with Toeplitz matrices in which our algorithm finds the correct solution within 1-2 iterations to a satisfactory accuracy. These experiments are presented in the last section.

3. Toeplitz Matrices

We would like to show that each iteration step can be computed relatively fast for Toeplitz matrices E. Let us discuss the algorithms that can be applied to each step.

For the underdetermined system (*) we do the following steps:

Step 1. $M = I + X^{(k-1)}(X^{(k-1)})^T$
Step 2. Solve $My^{(k)} = s^{(k-1)}$
Step 3. $\alpha^{(k)} = (X^{(k-1)})^T y^{(k)}$

Matrix $M \in \mathcal{R}^{m\times m}$ in Step 1 is banded $(2n-1)$-diagonal symmetric positive definite and Toeplitz. For example, for $m=5, n=3$ we have

$$M = \begin{pmatrix} 1+\sum_{i=1}^3 x_i^2 & x_1x_2+x_2x_3 & x_1x_3 & 0 & 0 \\ x_1x_2+x_2x_3 & 1+\sum_{i=1}^3 x_i^2 & x_1x_2+x_2x_3 & x_1x_3 & 0 \\ x_1x_3 & x_1x_2+x_2x_3 & 1+\sum_{i=1}^3 x_i^2 & x_1x_2+x_2x_3 & x_1x_3 \\ 0 & x_1x_3 & x_1x_2+x_2x_3 & 1+\sum_{i=1}^3 x_i^2 & x_1x_2+x_2x_3 \\ 0 & 0 & x_1x_3 & x_1x_2+x_2x_3 & 1+\sum_{i=1}^3 x_i^2 \end{pmatrix},$$

for some vector $x \in \mathcal{R}^3$. So, M is fully defined by its first row $m^{(1)}$. By direct calculation it can be checked that

$$m^{(1)} = (1+\sum_{i=1}^{n}(x_i^{(k-1)})^2,\ \sum_{i=1}^{n-1} x_i^{(k-1)}x_{i+1}^{(k-1)},\ \sum_{i=1}^{n-2} x_i^{(k-1)}x_{i+2}^{(k-1)}, \dots, x_1^{(k-1)}x_n^{(k-1)},\ 0,\dots,0).$$

The entries of $m^{(1)}$ are defined by the following matrix-vector product:

$$\begin{pmatrix} x_n^{(k-1)} & \cdots & x_1^{(k-1)} \\ & \ddots & \vdots \\ 0 & & x_n^{(k-1)} \end{pmatrix} \begin{pmatrix} x_n^{(k-1)} \\ \vdots \\ x_1^{(k-1)} \end{pmatrix},$$

in which the matrix is Toeplitz. So, this multiplication can be done with $O(n\log_2 n)$ flops by the FFT. If n is small then direct multiplication should be preferred because it would be faster. In fact, we do not need to form the matrix M explicitly when we use an iterative method to solve the symmetric positive definite Toeplitz system at Step 2.

At Step 2 we have to solve a linear system with the matrix M. As we noticed, M is s.p.d., and its smallest eigenvalue is no less than 1. Here we have several possibilities:

- direct solution by some banded solver: $(1/3)mn^2 + O(mn)$ flops;
- iterative solution: $O(m \log_2 m)N_{it}$ flops regardless of the preconditionning, where N_{it} is the number of iterative steps;
- direct solution by a super-fast Toeplitz solver (e. g. [1]): $8m \log_2^2 m + O(m)$ flops.

As far as M has very nice properties we would recommend the iterative solution, where an appropriate preconditioner (e. g. circulant [8]) can be applied as well.

Finally, Step 3 is a matrix-vector multiplication with the Toeplitz matrix $(X^{(k-1)})^T$, and can be done with $O((m+n)\log_2(m+n))$ flops.

Unfortunately, the overdetermined system (**) can not be solved efficiently as for general error matrices E. The reason is that matrix $E^{(k)}$ is not of low rank, in general. So, we have to solve it by some fast method. It can be done by the method proposed in [11] with $4mn + O(n^2)$ flops.

The computation of $s^{(k)}$ and $r^{(k)}$ involves products with Toeplitz matrices, so, we need $O((m+n)\log_2(m+n))$ flops. Summarizing, the total number of flops per iteration step is

$$4mn + O((m+n)\log_2(m+n)). \tag{17}$$

Let us also note that the memory we need is just for a few vectors of length not greater than $m+n$.

We can propose also another version of the algorithm which is about twice as fast but needs more memory. The only difference in this version is the solution of the LS problem (**). Let us suppose that at the initial step of Algorithm 4 we compute the QR factorization of matrix A, i. e. $A = QR$. Then we use the pseudo-inverse A^+ as a preconditioner. The preconditioned system looks as follows:

$$(I + R^{-1}Q^T E^{(k)})x = x^{(0)}.$$

Now the following iteration can be applied:

$$z^{(i)} = x^{(0)} - R^{-1}Q^T E^{(k)} z^{(i-1)}, \quad z^{(0)} = x^{(k-1)}. \tag{18}$$

The most time consuming operation in (18) is the matrix vector multiplication with the matrix Q^T. It needs $2mn + O(n)$ flops. Then the total flop count per iteration step of Algorithm 4 becomes

$$2mn + O((m+n)\log_2(m+n)). \tag{19}$$

This is asymptotically twice less but needs storage for the matrices Q and R.

For comparison we will mention the flop count of one iteration step for the STLS method in [15]. The flop count is not given in that reference but in the preceding technical report [14]. Asymptotically it is $O(mn^2 + m^2)$ which is clearly more than in our method (see (17) and (19)).

4. Numerical Experiments

All the examples are with Toeplitz matrices done in Matlab. The first two are taken from [15], and the second two from [12]. The stopping criterion in all the examples is

$$\left(\left\| \begin{matrix} r^{(k-1)} \\ \alpha^{(k-1)} \end{matrix} \right\|_2 - \left\| \begin{matrix} r^{(k)} \\ \alpha^{(k)} \end{matrix} \right\|_2 \right) / \left\| \begin{matrix} r^{(k)} \\ \alpha^{(k)} \end{matrix} \right\|_2 < 0.1,$$

i. e. we find the norm of the error with one correct digit approximately. This is enough because the next iterations will not improve the error essentially. In all the examples we give the norm of the error

$$ERRNORM = \left\| \begin{matrix} r^{(k)} \\ \alpha^{(k)} \end{matrix} \right\|_2$$

from the last iteration.

Example 1. Here $m = 6, n = 4$. The first column and row of the Toeplitz matrix A are:

$$\text{col} = [-3\ 7\ 10\ -1\ 0\ 0], \quad \text{row} = [-3\ 0\ 0\ 0].$$

The right hand side is:

$$b = [-12\ 25\ 62\ -59\ 16\ 100]^T.$$

For this example we have $ERRNORM$ = 6.58E-2, and

$$x = [4.0290\ 0.9056\ -5.0122\ 9.5310]^T.$$

Example 2. The matrix is the same as in the previous example. The only change is in the right hand side:

$$b = [-12\ 25\ 62\ -59\ 9\ 122]^T.$$

We have $ERRNORM$ = 6.62E-1, and

$$x = [3.4755\ 1.7893\ -6.3365\ 11.1582]^T.$$

Let us note that in both examples the solution is slightly different from the one presented in [15] but the error norm computed here is less than the corresponding error in [15].

Example 3. The matrix A and the right hand side b are as follows:

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 2 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 \end{pmatrix} \in \mathcal{R}^{n\times(n-1)},$$

$$b = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ n-1 \end{pmatrix} + e \in \mathcal{R}^{n-1},$$

where e is randomly generated and scaled so that $\|e\|_2 = 0.01\|b\|_2$. The results are presented in Table 1.

Table 1: The error norms for Example 3 for different n.

n	10	100	200	300
$ERRNORM$	7.18E-1	6.08E-2	1.46E-2	7.40E-3

Example 4. The first column and row of the Toeplitz matrix A are given as follows:

$$\text{col: } a_{i,1} = \begin{cases} \frac{1}{\sqrt{2\pi\alpha^2}}\exp\left(\frac{-(\omega-i+1)^2}{2\alpha^2}\right), & i = 1, 2, \ldots, 2\omega+1, \\ 0, & \text{otherwise}, \end{cases}$$

$$\text{row} = [a_{11}\ 0 \ldots 0],$$

where $\alpha = 1.25$, and $\omega = 8$. The right hand side is:

$$b = [1 \ldots 1]^T + e,$$

where e is randomly generated and scaled so that $\|e\|_2 = 0.01\|b\|_2$. The results are presented in Table 2.

Table 2: The error norms for Example 4 for different n.

n	64	100	200	500
$ERRNORM$	4.03E-1	3.94E-1	3.87E-1	3.84E-1

In all the examples the number of iterations was 2. This shows that the algorithm can be fast enough (especially if some fast, or super-fast, methods are applied for the basic iteration step).

Acknowledgment. We want to thank Jim Nagy for letting us use his MATLAB code for Toeplitz LS problems. This code is used in our experiments. We also like to give our sincere thanks to Professor Åke Börck for his very useful discussions and comments which improve the quality of the paper. The first author likes to thank Professor Jin Yun Yuan for his kind hospitality during his visit at the Federal University of Paraná, Brazil in 1997, and for introducing him to the topic. The second author likes to give his sincere thanks to Professor Gene Golub for his kind hospitality when he was visiting Stanford University, USA in 1998, and also to Professor Robert Plemmons for his helpful discussion and comments when he was visiting Wake Forest University, USA in 1997 and Spring, 1999. Finally, we thank referees for their helpful comments.

References

[1] G. Ammar and W. B.. Gragg, Superfast solution of real positive definite Toeplitz systems, *SIAM J. Matrix Anal. Appl.*, **9** (1988), 61-76.

[2] Å. Björck, Numerical Methods for Least Squares Problem, SIAM, Philadelphia, 1996.

[3] Å. Björck, Newton and Rayleigh quotient methods for total least squares problems, in: Recent Advances in Total Least Squares and Errors-in-Variables Modeling, SIAM, Philadelphia, 1997.

[4] R.H. Chan, J.G. Nagy and R.J. Plemmons, Displacement preconditioner for Toeplitz least squares iterations, *Electron. Trans. Numer. Anal.*, **2** (1994), 44-56.

[5] R.H. Chan, J.G. Nagy and R.J. Plemmons, FFT-based preconditioner for Toeplitz-block least squares problems, *SIAM J. Numer. Anal.*, **30** (1993), 1740-1768.

[6] R.H. Chan, J.G. Nagy, and R.J. Plemmons, Circulant preconditioned Toeplitz least squares iterations, *SIAM J. Matrix Anal. Appl.*, **15** (1994), 80-97.

[7] R..H. Chan, M.K. Ng and R.J. Plemmons, Generalization of Strang's preconditioner with applications to Toeplitz least squares problems, *Numer. Linear Algebra Appl.*, **3** (1996), 45-64.

[8] R. H. Chan and M. K. Ng, Conjugate gradient methods for Toeplitz systems, *SIAM Review*, **38** (1996), 427–482.

[9] G. H. Golub and C. F. van Loan, An analysis of the total least squares problem, *SIAM J. Numer. Anal.*, **17** (1980), 883-893.

[10] G. H. Golub and C. F. van Loan, Matrix Computations, 3rd ed., John Hopkins University Press, Baltimore, 1996.

[11] J. G. Nagy, Fast Inverse QR Factorization for Toeplitz Matrices, *SIAM J. Sci. Stat. Comput.*, **14** (1993), 1174-1193.

[12] J. Kamm and J. G. Nagy, A total least squares method for Toeplitz systems of equations, *BIT*, **38** (1998), 560-582.

[13] M.K. Ng and R.J. Plemmons, New Approach to Constrained Total Least Squares Image Restoration, Technical Report, Wake Forest University, Winston-Salem, NC, March, 1999.

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

[15] J. B. Rosen, H. Park, and J. Glick, Total least norm formulation and solution for structured problems, *SIAM J. Matrix Anal. Appl.*, **17** (1996), 110-126.

[16] D. B. Szyld, Criteria for combining inverse and Rayleigh quotient iteration, *SIAM J. Numer. Anal.*, **25** (1988), 1369-1375.

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