[J. Res. Natl. Inst. Stand. Technol. **111**, 113-119 (2006)]

Fast Algorithms for Structured Least Squares and Total Least Squares Problems

Volume 111 Number 2 March-April 2006

Anoop Kalsi

Applied Mathematics Program,
University of Maryland,
College Park, MD 20742

and

Dianne P. O'Leary

Dept. of Computer Science and
Institute for Advanced
Computer Studies,
University of Maryland,
College Park, MD 20742

and

National Institute of Standards
and Technology,
Gaithersburg, MD 20899-8910

oleary@cs.umd.edu

We consider the problem of solving least squares problems involving a matrix $\boldsymbol{M}$ of small displacement rank with respect to two matrices $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$. We develop formulas for the generators of the matrix $\boldsymbol{M}^H\boldsymbol{M}$ in terms of the generators of $\boldsymbol{M}$ and show that the Cholesky factorization of the matrix $\boldsymbol{M}^H\boldsymbol{M}$ can be computed quickly if $\boldsymbol{Z}_1$ is close to unitary and $\boldsymbol{Z}_2$ is triangular and nilpotent. These conditions are satisfied for several classes of matrices, including Toeplitz, block Toeplitz, Hankel, and block Hankel, and for matrices whose blocks have such structure. Fast Cholesky factorization enables fast solution of least squares problems, total least squares problems, and regularized total least squares problems involving these classes of matrices.

Key words: block Toeplitz matrix; displacement rank; errors in variables method; image deblurring; structured total least squares; Tikhonov regularization; total least squares.

Accepted: January 15, 2005

Available online: http://www.nist.gov/jres

1. Introduction

In [6], Mastronardi, Lemmerling, and Van Huffel present an algorithm for solving fast structured total least squares problems of the form

$$\min_{\boldsymbol{E},\boldsymbol{\beta},\boldsymbol{x}} \left\| \begin{bmatrix} \boldsymbol{E} & \boldsymbol{\beta} \end{bmatrix} \right\|_F^2 \qquad (1)$$

subject to the constraints

$$(\boldsymbol{A}+\boldsymbol{E})\boldsymbol{x} = \boldsymbol{y}+\boldsymbol{\beta}$$

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

In this paper, we work through the details of a small generalization of this algorithm. We consider the same problem (1), but under the constraint that $\boldsymbol{A}$ and $\boldsymbol{E}$ have small displacement rank relative to some matrices $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$. Choosing these two matrices to be shift-down matrices and the rank to be two gives the Toeplitz constraint considered by [6], but we will be interested in other cases as well. See [3] for a discussion of displacement structure.

We also consider fast solution of the problem under the additional constraint that the norm of the solution vector $\boldsymbol{x}$ is specified, a problem posed in Pruessner and O'Leary [10]. This corresponds to a Tikhonov regularization of our structured total least squares problem and results in a fast solution algorithm for the problem considered in [8, 9, 10, 7].

The core of the algorithm in [6], based on a more general algorithm of [11], relies on two results: the representation of the generators for the matrix $\boldsymbol{A}^H\boldsymbol{A}$ that appears in the normal equations when $\boldsymbol{A}$ is Toeplitz, and then a fast factorization of a matrix derived from these generators. So we begin in Sec. 2 with a construction of the generators for $\boldsymbol{M}^H\boldsymbol{M}$ when $\boldsymbol{M}$ is any matrix of small displacement rank and $\boldsymbol{Z}_1$ is close to unitary. In Sec. 3 we show that it is inexpensive to form a Cholesky factorization of $\boldsymbol{M}^H\boldsymbol{M}$ whenever $\boldsymbol{Z}_2$ is triangular and nilpotent. Section 4 concerns the application of this algorithm, and we conclude in Sec. 5.

The results in this work are derived from [4, 5].

2. Generators of M^HM

One way to solve a least squares problem

$$\min_{x} \| \boldsymbol{Mx} - \boldsymbol{b} \|_2$$

involving a matrix $\boldsymbol{M}$ of full column rank is to use the normal equation formulation; we factor $\boldsymbol{M}^H\boldsymbol{M} = \boldsymbol{LL}^H$ where $\boldsymbol{L}$ is a square lower triangular matrix and then solve the linear system

$$\boldsymbol{LL}^H\boldsymbol{x} = \boldsymbol{M}^H\boldsymbol{b} \ . \tag{2}$$

Thus, we can solve this problem fast if we can compute a Cholesky factorization of $\boldsymbol{M}^H\boldsymbol{M}$ fast. To do this, we will first derive a generator for the matrix $\boldsymbol{M}^H\boldsymbol{M}$ when $\boldsymbol{M} \in C^{m\times n}$ $(m \ge n)$ has low displacement rank.

Suppose that $\boldsymbol{M}$ has low displacement rank relative to the matrices $\boldsymbol{Z}_1 \in C^{m\times m}$ and $\boldsymbol{Z}_2 \in C^{n\times n}$, which means that if we define

$$\boldsymbol{N} \equiv \boldsymbol{M} - \boldsymbol{Z}_1\boldsymbol{M}\boldsymbol{Z}_2^H \ ,$$

then rank ($\boldsymbol{N}$), which we denote by ρ_1, is small relative to n.

Suppose

$$\tilde{\boldsymbol{Z}} = \boldsymbol{Z}_1 + \boldsymbol{W}$$

is a unitary matrix ($\tilde{\boldsymbol{Z}}^H\tilde{\boldsymbol{Z}} = \boldsymbol{I}$), where $\boldsymbol{W}$ has rank ρ_2, also assumed to be small. For example, if $\boldsymbol{E}$ is Toeplitz, let $\boldsymbol{Z}_1$ be the shift-down matrix with ones on its subdiagonal and zeros elsewhere, and then $\boldsymbol{W}$ is the matrix with a one in the last position of row 1.

Then $\boldsymbol{M}^H\boldsymbol{M}$ also has low displacement rank relative to $\boldsymbol{Z}_2$, as we can see from the identity

$$\begin{aligned}
&\boldsymbol{M}^H\boldsymbol{M} - \boldsymbol{Z}_2\boldsymbol{M}^H\boldsymbol{M}\boldsymbol{Z}_2^H = \boldsymbol{M}^H\boldsymbol{M} - \boldsymbol{Z}_2\boldsymbol{M}^H\tilde{\boldsymbol{Z}}^H\tilde{\boldsymbol{Z}}\boldsymbol{M}\boldsymbol{Z}_2^H\\
&= \boldsymbol{M}^H\boldsymbol{M} - (\boldsymbol{M} - \boldsymbol{N} + \boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H)^H(\boldsymbol{M} - \boldsymbol{N} + \boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H)\\
&= (\boldsymbol{N} - \boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H)^H(\boldsymbol{M} - \boldsymbol{N} + \boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H) + \boldsymbol{M}^H(\boldsymbol{N} - \boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H).
\end{aligned}$$

Theorem 1. If the rank of $\boldsymbol{N} \equiv \boldsymbol{M} - \boldsymbol{Z}_1\boldsymbol{M}\boldsymbol{Z}_2^H$ is ρ_1 and if the unitary matrix $\tilde{\boldsymbol{Z}}$ is equal to $\boldsymbol{Z}_1 + \boldsymbol{W}$ where $\boldsymbol{W}$ has rank ρ_2, then

$$\begin{aligned}
&\boldsymbol{M}^H\boldsymbol{M} - \boldsymbol{Z}_2\boldsymbol{M}^H\boldsymbol{M}\boldsymbol{Z}_2^H =\\
&\quad -\boldsymbol{N}^H\boldsymbol{N} + \boldsymbol{N}^H(\boldsymbol{M} + \boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H) + (\boldsymbol{M}^H + (\boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H)^H)\boldsymbol{N}\\
&\quad -(\boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H)^H(\boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H) - \boldsymbol{M}^H(\boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H) - (\boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H)^H\boldsymbol{M}
\end{aligned}$$

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

Proof: The equation in the statement of the theorem is a regrouping of the terms in the previous equation. The rank of $\boldsymbol{N} - \boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H$ is at most the rank of $\boldsymbol{N}$ plus the rank of $\boldsymbol{W}$, so the rank of the sum in that equation is at most $2(\rho_1 + \rho_2)$. []

An alternate proof of this theorem can be derived by noting that displacement rank is preserved by taking a Schur complement [3, Lemma 1.5.2], and $\boldsymbol{M}^H\boldsymbol{M}$ is the Schur complement of

$$\begin{bmatrix} -\boldsymbol{I} & \boldsymbol{M} \\ \boldsymbol{M}^H & \boldsymbol{0} \end{bmatrix}$$

which has displacement rank at most $2(\rho_1 + \rho_2)$ relative to the matrix

$$\begin{bmatrix} \boldsymbol{Z}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_2 \end{bmatrix}.$$

The bound $2(\rho_1 + \rho_2)$ can be an overestimate, since the $\boldsymbol{N}$ and $\boldsymbol{W}$ terms may interact. For example, for a Toeplitz matrix with rank computed relative to the shift-down matrices, the bound yields 6 while the actual rank is 4. If we use circular shifts, though, both the bound and the actual rank are 4.

Using the identity

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

we can easily symmetrize the generators.

3. Determining a Factorization of M^HM From the Generators

Theorem 1 tells us how to determine $\rho = 2\,(\rho_1 + \rho_2)$ vectors $\boldsymbol{h}_i$ so that

$$\boldsymbol{M}^H\boldsymbol{M} - \boldsymbol{Z}_2\boldsymbol{M}^H\boldsymbol{M}\boldsymbol{Z}_2^H = \sum_{i=1}^{\rho} s_i\boldsymbol{h}_i\boldsymbol{h}_i^H \tag{3}$$

where s_i equals plus or minus 1. When $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ are shift-down matrices, it has been shown [6, 1, 2] that this implies that

$$\boldsymbol{M}^H\boldsymbol{M} = \sum_{i=1}^{\rho} s_i \boldsymbol{L}(\boldsymbol{h}_i)\boldsymbol{L}(\boldsymbol{h}_i)^H$$

$$= [\boldsymbol{L}(\boldsymbol{h}_1) \ldots \boldsymbol{L}(\boldsymbol{h}_\rho)]\boldsymbol{S}\begin{bmatrix} \boldsymbol{L}(\boldsymbol{h}_1)^H \\ \vdots \\ \boldsymbol{L}(\boldsymbol{h}_\rho)^H \end{bmatrix}$$

where $\boldsymbol{S}$ = diag($\boldsymbol{S}_i$), $\boldsymbol{S}_i$ = diag(s_i) ($n \times n$), and $\boldsymbol{L}(\boldsymbol{h}_i)$ is the lower triangular Toeplitz matrix with first column equal to $\boldsymbol{h}_i$. We now generalize this result somewhat.

3.1 Triangular Factors of Structured Matrices

Theorem 2.

If $\boldsymbol{Z}_1$ is nilpotent, then

$$\boldsymbol{A} - \boldsymbol{Z}_1\boldsymbol{A}\boldsymbol{Z}_2^H = \boldsymbol{g}\boldsymbol{h}^H$$

and only if

$$\boldsymbol{A} = \boldsymbol{L}_1(\boldsymbol{g})\boldsymbol{L}_2^H(\boldsymbol{h})$$

where

$$\boldsymbol{L}_i(\boldsymbol{x}) = [\boldsymbol{x} \quad \boldsymbol{Z}_i\boldsymbol{x} \quad \ldots \quad \boldsymbol{Z}_i^{n-1}\boldsymbol{x}].$$

Proof: Suppose $\boldsymbol{A} = \boldsymbol{L}_1(\boldsymbol{g})\boldsymbol{L}_2^H(\boldsymbol{h})$. Observe that

$$\boldsymbol{L}_1(\boldsymbol{g})\boldsymbol{L}_2^H(h) = [\boldsymbol{g} \quad \boldsymbol{Z}_1\boldsymbol{g} \quad \ldots \quad \boldsymbol{Z}_1^{n-1}\boldsymbol{g}]\begin{bmatrix} \boldsymbol{h}^H \\ \boldsymbol{h}^H(\boldsymbol{Z}_2^H) \\ \vdots \\ \boldsymbol{h}^H(\boldsymbol{Z}_2^H)^{n-1} \end{bmatrix}$$

$$= \sum_{j=0}^{n-1} \boldsymbol{Z}_1^j \boldsymbol{g}\boldsymbol{h}^H \boldsymbol{Z}_2^j$$

and

$$\boldsymbol{Z}_1\boldsymbol{L}_1(\boldsymbol{g})\boldsymbol{L}_2^H(\boldsymbol{h})\boldsymbol{Z}_2^H = \sum_{j=0}^{n-1} \boldsymbol{Z}_1^{j+1}\boldsymbol{g}\boldsymbol{h}^H\boldsymbol{Z}_2^{j+1}$$

so, since $\boldsymbol{Z}_1^n = \boldsymbol{0}$, we conclude that

$$\boldsymbol{L}_1(\boldsymbol{g})\boldsymbol{L}_2^H(\boldsymbol{h}) - \boldsymbol{Z}_1\boldsymbol{L}_1(\boldsymbol{g})\boldsymbol{L}_2^H(\boldsymbol{h})\boldsymbol{Z}_2^H = \boldsymbol{g}\boldsymbol{h}^H.$$

To prove the converse, suppose $\boldsymbol{A}–\boldsymbol{Z}_1\boldsymbol{A}\boldsymbol{Z}_2^H = \boldsymbol{g}\boldsymbol{h}^H$. Then, since

$$\boldsymbol{g}\boldsymbol{h}^H = \boldsymbol{L}_1(\boldsymbol{g})\boldsymbol{L}_2^H(\boldsymbol{h}) - \boldsymbol{Z}_1\boldsymbol{L}_1(\boldsymbol{g})\boldsymbol{L}_2^H(\boldsymbol{h})\boldsymbol{Z}_2^H,$$

we conclude that if $\boldsymbol{E} = \boldsymbol{A} - \boldsymbol{L}_1(\boldsymbol{g})\, \boldsymbol{L}_2^H$ (**h**), then

$$\boldsymbol{E} = \boldsymbol{Z}_1\boldsymbol{E}\boldsymbol{Z}_2^H.$$

Now since $\boldsymbol{Z}_1$ is nilpotent, $\boldsymbol{Z}_1^p = \boldsymbol{0}$ for some $p \le n$. Therefore, $\boldsymbol{Z}_1^{p-1}\boldsymbol{E} = \boldsymbol{Z}_1^p\boldsymbol{E}\boldsymbol{Z}_2^H = \boldsymbol{0}$, and working backward in powers of $\boldsymbol{Z}_1$, we see that $\boldsymbol{Z}_1^0\boldsymbol{E} = \boldsymbol{Z}_1\boldsymbol{E}\boldsymbol{Z}_2^H = \boldsymbol{0}$, so $\boldsymbol{A} = \boldsymbol{L}_1(\boldsymbol{g})\boldsymbol{L}_2^H(\boldsymbol{h})$. []

The following corollary can be proved by finite induction.

Corollary 1: If $\boldsymbol{Z}_1$ is nilpotent, then

$$\boldsymbol{A} - \boldsymbol{Z}_1\boldsymbol{A}\boldsymbol{Z}_2^H = \sum_{i=1}^{\rho} \boldsymbol{g}_i\boldsymbol{h}_i^H$$

if and only if

$$\boldsymbol{A} = \sum_{i=1}^{\rho} \boldsymbol{L}_1(\boldsymbol{g}_i)\boldsymbol{L}_2^H(\boldsymbol{h}_i).$$

3.2 Triangular Factors of M^HM

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

$$\boldsymbol{M}^H\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^H,$$

so we need to reduce the matrix

$$\begin{bmatrix} \boldsymbol{L}_2(\boldsymbol{h}_1)^H \\ \vdots \\ \boldsymbol{L}_2(\boldsymbol{h}_\rho)^H \end{bmatrix}$$

to upper triangular form, where the vectors $\boldsymbol{h}_i$ are defined in (3).

If $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ are shift-down matrices, then [6] shows how to do this reduction fast. Using Corollary 1, we will see that this can be done fast whenever $\boldsymbol{Z}_2$ is triangular and nilpotent. We present the algorithm for the case $\boldsymbol{Z}_2$ lower triangular; the upper triangular case is analogous but works with the columns in reverse order.

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

$$\hat{\boldsymbol{L}} \equiv \begin{bmatrix} \boldsymbol{L}_2(\boldsymbol{h}_1)^H \\ \vdots \\ \boldsymbol{L}_2(\boldsymbol{h}_\rho)^H \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1^H \\ \boldsymbol{h}_1^H\boldsymbol{Z}_2^H \\ \cdots \\ \boldsymbol{h}_1^H(\boldsymbol{Z}_2^H)^n \\ \vdots \\ \boldsymbol{h}_\rho^H \\ \boldsymbol{h}_\rho^H\boldsymbol{Z}_2^H \\ \cdots \\ \boldsymbol{h}_\rho^H(\boldsymbol{Z}_2^H)^n \end{bmatrix}.$$

Suppose we determine a rotation between the first row $\boldsymbol{h}_1^H$ and row $n + 1$, which contains $\boldsymbol{h}_2^H$, to zero the first element of $\boldsymbol{h}_2^H$. The same rotation between $\boldsymbol{h}_1^H(\boldsymbol{Z}_2^H)^j$ and $\boldsymbol{h}_2^H(\boldsymbol{Z}_2^H)^j$ ($j = 1, \ldots, n - 1$) also zeroes the first element of $\boldsymbol{h}_2^H(\boldsymbol{Z}_2^H)^j$ since $\boldsymbol{Z}_2^H$ is upper triangular. Therefore, by introducing one zero into our matrix, we have implicitly introduced $n - 1$ more, so we can put zeroes below the main diagonal in column 1 by using only $\rho - 1$ rotations, independent of the size of n.

We then use the resulting second row, equal to the first row postmultiplied by $\boldsymbol{Z}_2^H$, to zero the second element of row $n + 1$. Again this implicitly introduces additional zeros, $n - 2$ of them, and we complete the operations on column 2 by using $\rho - 1$ rotations.

If we repeat this for each column, we accomplish our reduction. Let $\boldsymbol{H}$ be the $\rho \times n$ matrix whose rows are $\boldsymbol{h}_i^H$. We can thus reduce $\hat{\boldsymbol{L}}$ to upper triangular form just by operating on the matrix $\boldsymbol{H}$.

We design our algorithm to use Givens rotations as often as possible, minimizing the number of hyperbolic rotations in order to preserve stability. A Givens rotation can be used between row i and row j whenever s_i and s_j (see equation (3)) have the same sign; if they have different signs, then we must use a hyperbolic rotation. We'll assume that we have ordered the generators so that the first $\hat{\rho}$ rows of $\boldsymbol{H}$ have $s_i = 1$ and the remaining ones have $s_i = -1$.

Algorithm Reduce ($\boldsymbol{H}$)[3]

```
For $j = 1, \ldots, n$,

    For $i = 2, \ldots, \hat{\rho}$,

        If $h_{ij}$ is nonzero, then

            zero it by a Givens rotation between
            row 1 and row $i$;

    end for
    For $i = \hat{\rho} + 2, \ldots, \rho$,
        If $h_{ij}$ is nonzero, then

            zero it by a Givens rotation between
            row $\hat{\rho} + 1$ and row $i$;

    end for

    If $h_{\hat{\rho}+1,j}$ is nonzero, then
        zero it by a stabilized hyperbolic rotation
        between row 1 and row $\hat{\rho} + 1$;

    Then the $j$th row of $\boldsymbol{L}^H$ is $\boldsymbol{h}_1^H$, the first row
    of the current $\boldsymbol{H}$ matrix.

    Replace the first row of $\boldsymbol{H}$ by $\boldsymbol{h}_1^H \boldsymbol{Z}_2^H$ to
    form the pivot row for the next value of $j$.

end for
```

The cost of this reduction is at most $O(\rho n^2)$, ignoring sparsity, plus the cost of the multiplications by $\boldsymbol{Z}_2$. Sometimes sparsity can reduce this cost significantly [7]. Without exploiting the structure of $\hat{\boldsymbol{L}}$ the cost would be $O(\rho n^3)$. Once the factors $\boldsymbol{LL}^H$ are computed, they can then be used to solve (2).

4. Some Applications

Our fast algorithm enables us to solve least squares, total least squares, and regularized total least squares problems involving matrices for which $\boldsymbol{Z}_1$ is close to unitary and $\boldsymbol{Z}_2$ is triangular and nilpotent. This includes several important classes of structured matrices.

4.1 Solving Least Squares Problems

Consider first the least squares problem

$$\min_{\boldsymbol{x}} \| \boldsymbol{Mx} - \boldsymbol{b} \|_2$$

where $\boldsymbol{M} \in C^{m \times n}$ has full column rank. We can use our algorithm if

- $\boldsymbol{M}$ is Toeplitz. Then $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ are the shift-down matrices of appropriate size and the displacement rank is 2.

- $\boldsymbol{M}$ is block Toeplitz:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_\delta & \boldsymbol{M}_{\delta-1} & \cdots & \boldsymbol{M}_1 \\ \boldsymbol{M}_{\delta+1} & \boldsymbol{M}_\delta & \cdots & \boldsymbol{M}_2 \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{M}_{\delta+\gamma-1} & \boldsymbol{M}_{\delta+\gamma-2} & \cdots & \boldsymbol{M}_\delta \end{bmatrix},$$

where $\boldsymbol{M}_\delta$ has dimension $m / \gamma \times n / \delta$. Then choosing $\boldsymbol{Z}_1 \in C^{m \times m}$ as

[3] There is an analogous algorithm, FTriang, in [6], for the special case in which $\boldsymbol{M}$ is Toeplitz, but it has some typographical errors. In the statement following "if i<m", g3 on the left-hand side should be g4. In 12 places on p. 552, "m+n" should be "mn1". Also, the numbering of the phases of the computation is off by one compared with the description in the paper ("Initialization" should be "Phase 1", etc.).

$$\boldsymbol{Z}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \boldsymbol{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{I} & \mathbf{0} \end{bmatrix},$$

where each block has dimension $m / \gamma \times m / \gamma$, and choosing $\boldsymbol{Z}_2 \in C^{m \times n}$ similarly but with blocks of dimension $n / \delta \times n / \delta$ gives a displacement rank of $m / \gamma + n / \delta$.

- $\boldsymbol{M}$ has Toeplitz blocks:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} & \dots & \boldsymbol{M}_{1,\delta} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} & \dots & \boldsymbol{M}_{2,\delta} \\ \dots & \dots & \dots & \dots \\ \boldsymbol{M}_{\gamma,1} & \boldsymbol{M}_{\gamma,2} & \dots & \boldsymbol{M}_{\gamma,\delta} \end{bmatrix},$$

where each block $\boldsymbol{M}_{ij}$ is Toeplitz. Then choosing $\boldsymbol{Z}_1 \in C^{m \times m}$ as a block-diagonal matrix consisting of shift-down matrices of dimensions matching the column dimension of the diagonal blocks of $\boldsymbol{M}$ and choosing $\boldsymbol{Z}_2 \in C^{n \times n}$ similarly but with blocks matching the row dimension gives a displacement rank of $2(\gamma + \delta)$.

- $\boldsymbol{M}$ Hankel, block Hankel, or having Hankel blocks. These cases are analogous to those considered above.

4.2 Solving Structured Total Least Squares Problems

In some cases the matrix of the problem can be estimated only with error, and we need to determine not just the parameters $\boldsymbol{x}$ of the least squares fit but also the corrections to the operator. This problem can be formulated as

$$\min_{\boldsymbol{E},\boldsymbol{\beta},\boldsymbol{x}} \| [\boldsymbol{E} \quad \boldsymbol{\beta}] \|_F^2 \tag{4}$$

subject to the constraints

$$(\boldsymbol{A} + \boldsymbol{E})\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{\beta}$$

with $\boldsymbol{A}$ and $\boldsymbol{E}$ having the same structure.

Suppose that the matrix $\boldsymbol{E}$ can be specified by p parameters $\alpha_1, \dots, \alpha_p$. For example, if E is a Toeplitz matrix, then

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

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

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x}} \left\| \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\alpha} \end{bmatrix} \right\|_2^2 \tag{5}$$

where

$$\boldsymbol{\beta} = (\boldsymbol{A} + \boldsymbol{E})\boldsymbol{x} - \boldsymbol{y}.$$

Following [6], we have replaced the term $\| \boldsymbol{E} \|_F^2$ by $\boldsymbol{\alpha}^H \boldsymbol{\alpha}$, equivalent except for scaling of the entries α_ι^2.

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

$$\boldsymbol{Xa} = \boldsymbol{Ex}.$$

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

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

Following [11], we form a quadratic approximation to (5) by using linear approximations $\boldsymbol{\alpha} + \boldsymbol{\Delta\alpha}$ and $\boldsymbol{x} + \boldsymbol{\Delta x}$, resulting in

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

so that

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

If we minimize this with respect to $\boldsymbol{\Delta\alpha}$ and $\boldsymbol{\Delta x}$, then we can form a new approximation

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

to the solution of (5) and then repeat the procedure until convergence. As noted by [11], this is a Gauss-Newton algorithm applied to (5). Therefore, the main computational task is to solve linear least squares problems of the form

$$\min_{\boldsymbol{\Delta\alpha},\boldsymbol{\Delta x}} \left\| \boldsymbol{M} \begin{bmatrix} \boldsymbol{\Delta\alpha} \\ \boldsymbol{\Delta x} \end{bmatrix} + \begin{bmatrix} (\boldsymbol{A} + \boldsymbol{E})\boldsymbol{x} - \boldsymbol{y} \\ \boldsymbol{\alpha} \end{bmatrix} \right\|_2^2 \tag{6}$$

where

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

If $\boldsymbol{A}$ is in one of the classes considered in Sec. 4.1, then the matrix $\boldsymbol{M}$ has low displacement rank and we can solve the problem fast.

4.3 Solving Regularized Total Least Squares Problems

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

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

$$\| \boldsymbol{Cx} \| \leq u$$

where u is a given scalar and $\boldsymbol{C}$ is commonly chosen to be the identity matrix or a difference operator. If $\boldsymbol{C}$ has low displacement rank relative to $\boldsymbol{Z}_2$ and an appropriate $\hat{\boldsymbol{Z}}_1$, our algorithm can be easily modified to incorporate regularization. In this case, our problem (5) can be reformulated as

$$\min_{\alpha,\beta,x} \left\| \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\alpha} \\ \lambda \boldsymbol{Cx} \end{bmatrix} \right\|_2^2 \tag{7}$$

where $\boldsymbol{\beta} = (\boldsymbol{A} + \boldsymbol{E})\boldsymbol{x} - \boldsymbol{y}$ and λ, the regularization parameter, is the Lagrange multiplier for the new constraint. Using a derivation similar to that above, the linearization of (7) results in the following problem to be solved at each step of the iteration:

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

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

The displacement structure of this matrix is greatly simplified if $\boldsymbol{C}$ is upper triangular and $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ are shift-down matrices. As noted before, $\boldsymbol{W}$ is zero except for a one in the last position of the first row, and thus $\boldsymbol{WM}$ is zero except for a λ in the last position of the first row. Therefore, $\boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H = \boldsymbol{0}$, so, applying Theorem 1, we have the following result.

Theorem 3. If $\boldsymbol{C}$ is upper triangular and $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ are shift-down matrices, then

$$\boldsymbol{M}^H\boldsymbol{M} - \boldsymbol{Z}_2\boldsymbol{M}^H\boldsymbol{M}\boldsymbol{Z}_2^H = (\boldsymbol{M} - \boldsymbol{N}/2)^H\,\boldsymbol{N} + \boldsymbol{N}^H(\boldsymbol{M} - \boldsymbol{N}/2) \tag{8}$$

and has rank $2\rho_1$, where ρ_1 is the rank of $\boldsymbol{N}$. More generally, (8) holds whenever $\boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}_2^H = \boldsymbol{0}$.

5. Conclusions

We have derived the generators for $\boldsymbol{M}^H\boldsymbol{M}$ when $\boldsymbol{M}$ is any matrix of small displacement rank relative to $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$. We have shown that it is inexpensive to form a Cholesky factorization of $\boldsymbol{M}^H\boldsymbol{M}$ whenever $\boldsymbol{Z}_1$ is close to unitary and $\boldsymbol{Z}_2$ is triangular and nilpotent, and we have generalized this algorithm when a regularization constraint is to be applied.

Acknowledgments

We are grateful to Haw-ren Fang for helpful comments on the manuscript. This work was partially supported by the National Science Foundation under Grant CCR-0204084.

6. References

[1] J. Chun, T. Kailath, and H. Lev-Ari, Fast parallel algorithms for QR and triangular factorizations, SIAM J. Sci. Stat. Comput. **8**, 899-913 (1987).

[2] T. Kailath, S. Kung, and M. Morf, Displacement ranks of matrices and linear equations, J. Math. Anal. Appl. **68**, 395-407 (1979).

[3] Thomas Kailath, Displacement structure and array algorithms, in Fast Reliable Algorithms for Matrices with Structure, SIAM, Philadelphia (1999) pp. 1-56.

[4] Anoop Kalsi, Blind two-dimensional deconvolution using total least squares, M.A. Thesis, Mathematics Department, University of Maryland (2002).

[5] Anoop Kalsi and Dianne O'Leary, Algorithms for structured total least squares problems with applications to blind image deblurring, Technical report, Computer Science Department Report CS-TR-4390, University of Maryland, August 2002.

[6] N. Mastronardi, P. Lemmerling, and S. Van Huffel, Fast structured total least squares algorithm for solving the basic deconvolution algorithm, SIAM J. Matrix Anal. Appl. **22**, 533-553 (2000).

[7] Nicola Mastronardi, Phillip Lemmerling, Anoop Kalsi, Dianne O'Leary, and Sabine Van Huffel, Regularized structured total least squares algorithms for blind image deblurring, Linear Algebra and Its Appl. **391**, 203-221 (2004).

[8] V. Mesarović, N. Galatsanos, and A. Katsaggelos, Regularized constrained total least squares image restoration, IEEE Transactions on Image Processing **4**, 1096-1108 (1995).

[9] M. K. Ng, R. J. Plemmons, and F. Pimentel, A new approach to constrained total least squares image restoration, Linear Algebra and Its Appl. **316**, 237-258 (2000).

[10] Armin Pruessner and Dianne P. O'Leary, Blind deconvolution using a regularized structured total least norm algorithm, SIAM J. on Matrix Anal. Appl.2**4**, 1018-1037 (2003).

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

About the authors: *Dianne O'Leary is a professor of computer science at the University of Maryland and a mathematician in the NIST Mathematical and Computational Sciences Division. She has a B.S. from Purdue University and a Ph.D from Stanford University. She is the author of over 75 publications on numerical analysis and computational science and over 20 publications on education and mentoring. Anoop Kalsi completed her M.S. degree in the Applied Mathematics Program at the University of Maryland in 2002, and this work formed the basis for her thesis. She is now teaching in New Jersey. The National Institute of Standards and Technology is an agency of the Technology Admininstration, U.S. Department of Commerce.*