A Rank-Revealing Decomposition of Symmetric Toeplitz Matrices

Franklin T. Luk
Computer Science Department, Rensselaer Polytechnic Institute
Troy, New York 12180, USA

Sanzheng Qiao
Communications Research Laboratory, McMaster University
Hamilton, Ontario L8S 4K1, Canada

ABSTRACT

In signal and image processing, regularization often requires a rank-revealing decomposition of a symmetric Toeplitz matrix with a small rank deficiency. In this paper, we present an efficient factorization method that exploits symmetry as well as the rank and Toeplitz properties of the given matrix.

Keywords: Toeplitz matrix, regularization, symmetric rank-revealing decomposition.

1. INTRODUCTION

In signal and image processing applications [5], [6], a noisy and distorted signal vector $\hat{x}$ is given by

$$\hat{x} = Tx + w, \tag{1.1}$$

where x and w represent an unknown original signal vector and a noise vector, respectively, and T is a predetermined matrix describing the spread of signals. Often, T is real, symmetric, and Toeplitz, i.e., all entries on the same diagonal are equal. Thus, assuming the dimension of x is n, we have

$$T = \begin{pmatrix} t_1 & t_2 & \cdots & t_n \\ t_2 & t_1 & \cdots & t_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ t_n & t_{n-1} & \cdots & t_1 \end{pmatrix}. \tag{1.2}$$

That is, T is determined by an n-vector $(t_1, t_2, \cdots, t_n)^T$. To restore the original signal vector x from $\hat{x}$, we should 'invert' T. Very often (see, e.g., [1]), T is ill-conditioned and regularization is required. Assume that T has a small rank deficiency, i.e.,

$$\mathrm{rank}(T) = n - k, \tag{1.3}$$

where k denotes a small integer. One regularization method is the truncated singular value decomposition (TSVD) [4], which requires $O(n^3)$ floating-point operations (flops). In this paper, we present an $O(kn^2)$ method for computing a rank-revealing factorization of a symmetric Toeplitz matrix. Examples in the last section show that our new method restores the signal almost as accurately as the TSVD.

The paper is organized as follows. We discuss rank-revealing factorizations for general and symmetric matrices in Section 2. In Section 3, we sketch how to compute a triangular decomposition of a symmetric Toeplitz matrix and present a new computational scheme that avoids the use of complex arithmetic. In Section 4, we describe a rank-revealing procedure. Simulation results are given in Section 5.

2. RANK-REVEALING TECHNIQUES

In this section, we discuss cost-effective rank-revealing methods for matrices. In §2.1 we present the URV decomposition [7] for general matrices. To date, no one has shown how to exploit symmetry in this decomposition. We propose a possible extension in §2.2.

2.1. General Case

Consider the TSVD procedure. We compute the SVD:

$$T = U\Sigma V^T, \tag{2.1.1}$$

where U and V are orthogonal, and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. Since T is ill-conditioned, we select a tolerance τ to find k such that

$$\sigma_{n-k} > \tau \geq \sigma_{n-k+1}. \tag{2.1.2}$$

We may say that T has k 'small' singular values, or that T has a numerical rank of $n-k$. Denoting

$$\hat{\Sigma} = \mathrm{diag}(\sigma_1, \cdots, \sigma_{n-k}, 0, \cdots, 0),$$

we restore the original signal via $x = V\hat{\Sigma}^\dagger U^T \hat{x}$, where $\hat{\Sigma}^\dagger = \mathrm{diag}(\sigma_1^{-1}, \cdots, \sigma_{n-k}^{-1}, 0, \cdots, 0)$.

Although the TSVD is stable, it is expensive to compute. Since regularization does not require the diagonalization of T, we may pick the less costly URV decomposition:

$$T = URV^T, \tag{2.1.3}$$

where U and V are orthogonal, and R is upper triangular, viz.,

$$R = \begin{pmatrix} \bar{R} & E \\ 0 & G \end{pmatrix}. \tag{2.1.4}$$

The three submatrices $\bar{R}$, E and G possess the special properties that $||\bar{R}||$ is 'large', and $||E||$ and $||G||$ are 'small':

$$\sigma_{\min}(\bar{R}) \approx \sigma_{n-k}, \qquad ||E||_F^2 + ||G||_F^2 \approx \sigma_{n-k+1}^2 + \cdots + \sigma_n^2. \tag{2.1.5}$$

A regularized solution to (1.1) is obtained from

$$x = V \begin{pmatrix} \bar{R}^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T \hat{x}.$$

In [6], Nagy proposed an $O(bn^2)$ method for computing an approximate URV decomposition of a banded Toeplitz matrix with half-bandwidth b; the cost of his method appears to be independent of k.

2.2. Symmetric Case

Here, we wish to exploit the symmetry of T to save on storage and work. First, we compute an eigenvalue decomposition:

$$T = V\Sigma V^T, \tag{2.2.1}$$

where V is orthogonal, and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$ with $|\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_n| \geq 0$. We choose a tolerance τ to find k such that

$$|\sigma_{n-k}| > \tau \geq |\sigma_{n-k+1}|, \tag{2.2.2}$$

i.e., T has a numerical rank of $n-k$ based on τ. Let

$$\hat{\Sigma} = \mathrm{diag}(\sigma_1, \cdots, \sigma_{n-k}, 0, \cdots, 0).$$

As in §2.1 we restore the original signal using $x = V\hat{\Sigma}^{\dagger}V^T\hat{x}$.

Here, we propose the following generalization:

$$T = VSV^T, \tag{2.2.3}$$

where V is orthogonal and S symmetric – we call it a VSV decomposition. Partition the matrix S:

$$S \equiv \begin{pmatrix} \bar{S} & E \\ E^T & G \end{pmatrix}, \tag{2.2.4}$$

where the three submatrices $\bar{S}$, E and G possess similar 'norm' properties as in (2.1.5):

$$\sigma_{\min}(\bar{S}) \approx \sigma_{n-k}, \qquad \|E\|_F^2 + \|\mathrm{triu}(G)\|_F^2 \approx \sigma_{n-k+1}^2 + \cdots + \sigma_n^2, \tag{2.2.5}$$

and $\mathrm{triu}(G)$ denotes the upper triangular part of G. Note that (2.2.5) is not an obvious extension of (2.1.5) to the symmetric case. These three submatrices are generated by

$$\begin{pmatrix} \bar{S} & E \\ E^T & G \end{pmatrix} = \begin{pmatrix} J & F \\ 0 & K \end{pmatrix}^T \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \begin{pmatrix} J & F \\ 0 & K \end{pmatrix}, \tag{2.2.6}$$

where J and K are upper triangular, and D_1 and D_2 are *signature* matrices (diagonal matrices consisting of ± 1; see [3]). The original signal can be restored by a truncated VSV decomposition:

$$x = V \begin{pmatrix} \bar{S}^{-1} & 0 \\ 0 & 0 \end{pmatrix} V^T \hat{x}.$$

3. TRIANGULAR FACTORIZATION

In this section we discuss the fast triangular decomposition of a symmetric Toeplitz matrix. Our contribution is to show how complex arithmetic can be avoided when the given matrix is real.

3.1. *RDR* Decomposition

Consider the matrix T of (1.2). Assume that $t_1 > 0$; otherwise consider $-T$ (when $t_1 < 0$) or $T + I$ (when $t_1 = 0$). From [2] we get a displacement representation of T:

$$T = R_1^T R_1 - R_2^T R_2, \tag{3.1.1}$$

where R_1 and R_2 are upper triangular Toeplitz matrices whose first rows are given by $1/\sqrt{t_1}(t_1, t_2, ..., t_n)$ and $1/\sqrt{t_1}(0, t_2, ..., t_n)$, respectively. Rewriting (3.1.1) to get

$$T = (R_1^T \;\; R_2^T) \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}, \tag{3.1.2}$$

we use hyperbolic rotations to eliminate the matrix R_2 in (3.1.2) and obtain

$$T = (R^T \;\; 0) \begin{pmatrix} D & 0 \\ 0 & -\hat{D} \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix} = R^T D R, \tag{3.1.3}$$

where R is upper triangular, and D and $\hat{D}$ are signature matrices.

3.2. Avoiding Complex Arithmetic

Since T is real, it is important to restrict the computation to real arithmetic. Here, we need only the resultant matrices R and D, and so it is not necessary to explicitly construct the hyperbolic rotations. In this subsection, we present details on how to eliminate the matrix R_2 in (3.1.2) by applying hyperbolic rotations implicitly so that only real arithmetic is performed.

Given two real vectors a and b, and two scalars d_1, $d_2 = \pm 1$, consider the problem of finding real vectors $\hat{a}$ and $\hat{b}$, and scalars $\hat{d}_1$, $\hat{d}_2 = \pm 1$ so that

$$(a\ b) \begin{pmatrix} d_1 & 0 \\ 0 & -d_2 \end{pmatrix} \begin{pmatrix} a^T \\ b^T \end{pmatrix} = (\hat{a}\ \hat{b}) \begin{pmatrix} \hat{d}_1 & 0 \\ 0 & -\hat{d}_2 \end{pmatrix} \begin{pmatrix} \hat{a}^T \\ \hat{b}^T \end{pmatrix}, \tag{3.2.1}$$

and the first element of $\hat{b}$ is zero. Writing

$$(a\ b) \begin{pmatrix} d_1 & 0 \\ 0 & -d_2 \end{pmatrix} \begin{pmatrix} a^T \\ b^T \end{pmatrix} = (\sqrt{d_1}a\ \sqrt{d_2}b) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \sqrt{d_1}a^T \\ \sqrt{d_2}b^T \end{pmatrix},$$

we can find a hyperbolic rotation Y, defined by

$$Y = \begin{pmatrix} c & -s \\ -s & c \end{pmatrix}, \qquad \text{with} \quad c^2 - s^2 = 1,$$

to eliminate the first element of $\sqrt{d_2}b$, i.e.,

$$Y \begin{pmatrix} \sqrt{d_1}a^T \\ \sqrt{d_2}b^T \end{pmatrix} = \begin{pmatrix} \sqrt{\hat{d}_1}\hat{a}^T \\ \sqrt{\hat{d}_2}\hat{b}^T \end{pmatrix}.$$

Let α_1 and β_1 denote the first elements of a and b, respectively. Then Y is determined by

$$c = \frac{\sqrt{d_1}\alpha_1}{\sqrt{d_1\alpha_1^2 - d_2\beta_1^2}} \quad \text{and} \quad s = \frac{\sqrt{d_2}\beta_1}{\sqrt{d_1\alpha_1^2 - d_2\beta_1^2}}. \tag{3.2.2}$$

However, Y can be complex which is undesirable since the given data are real. We show how to maintain real arithmetic in the elimination procedure.

When $d_1\alpha_1^2 > d_2\beta_1^2$, we choose

$$\hat{c} = c/\sqrt{d_1} \quad \text{and} \quad \hat{s} = s/\sqrt{d_2}.$$

Then both

$$\hat{c} = \alpha_1/\sqrt{d_1\alpha_1^2 - d_2\beta_1^2} \quad \text{and} \quad \hat{s} = \beta_1/\sqrt{d_1\alpha_1^2 - d_2\beta_1^2}$$

are real. It follows that

$$\sqrt{\hat{d}_1}\,\hat{a} = c\sqrt{d_1}a - s\sqrt{d_2}b = d_1\hat{c}a - d_2\hat{s}b \tag{3.2.3}$$

and

$$\sqrt{\hat{d}_2}\,\hat{b} = -s\sqrt{d_1}a + c\sqrt{d_2}b = \sqrt{d_1d_2}(-\hat{s}a + \hat{c}b). \tag{3.2.4}$$

Since there are four unknowns in the two equations (3.2.3) and (3.2.4), we have the freedom to choose $\hat{d}_1$ and $\hat{d}_2$ to maintain real arithmetic. Let

$$\hat{d}_1 = 1 \quad \text{and} \quad \hat{d}_2 = d_1d_2.$$

Then the two quantities

$$\hat{a} = d_1\hat{c}a - d_2\hat{s}b \quad \text{and} \quad \hat{b} = -\hat{s}a + \hat{c}b$$

can be computed with real operations.

When $d_1\alpha_1^2 < d_2\beta_1^2$, we define

$$\hat{c} = \frac{\alpha_1}{\sqrt{d_2\beta_1^2 - d_1\alpha_1^2}} \quad \text{and} \quad \hat{s} = \frac{\beta_1}{\sqrt{d_2\beta_1^2 - d_1\alpha_1^2}}.$$

From (3.2.2) it follows that

$$c = \left(\frac{\sqrt{d_1}}{i}\right)\hat{c} \quad \text{and} \quad s = \left(\frac{\sqrt{d_2}}{i}\right)\hat{s},$$

where $i = \sqrt{-1}$. Thus, we get

$$\sqrt{\hat{d}_1}\hat{a} = \frac{1}{i}(d_1\hat{c}a - d_2\hat{s}b) \quad \text{and} \quad \sqrt{\hat{d}_2}\hat{b} = \frac{1}{i}\sqrt{d_1 d_2}(-\hat{s}a + \hat{c}b).$$

Choosing

$$\hat{d}_1 = -1 \quad \text{and} \quad \hat{d}_2 = -d_1 d_2,$$

we can compute $\hat{a}$, $\hat{b}$, $\hat{d}_1$, and $\hat{d}_2$ using real arithmetic as shown in the next subsection.

3.3. Quadratic Rotations

Summarizing the work in §3.2, we derive a procedure that we call *qrotation* (for *quadratic* rotation).

Procedure qrotation. Given vectors a, b, and scalars d_1, $d_2 = \pm 1$, this procedure computes vectors $\hat{a}$, $\hat{b}$, and scalars $\hat{d}_1$, $\hat{d}_2 = \pm 1$ in equation (3.2.1) so that the first element of $\hat{b}$ is zero.

check if $d_1\alpha_1^2 = d_2\beta_1^2$ or if $\beta_1 = 0$
if $|\alpha_1| > |\beta_1|$
  $t = d_1 - d_2(\beta_1/\alpha_1)^2$
  $\hat{t} = \sqrt{|t|}\ |\alpha_1|$
else
  $t = d_1(\alpha_1/\beta_1)^2 - d_2$
  $\hat{t} = \sqrt{|t|}\ |\beta_1|$
if $t > 0$
  $\hat{d}_1 = 1$
  $\hat{d}_2 = d_1 d_2$
else
  $\hat{d}_1 = -1$
  $\hat{d}_2 = -d_1 d_2$
$\hat{c} = \alpha_1/\hat{t}$
$\hat{s} = \beta_1/\hat{t}$
$\hat{a} = d_1\hat{c}a - d_2\hat{s}b$
$\hat{b} = -\hat{s}a + \hat{c}b$

The above procedure unifies circular and hyperbolic rotations so that it chooses either a circular rotation or a hyperbolic rotation depending on whether $d_1 d_2 = -1$ or $d_1 d_2 = 1$, respectively. We use *qrotation* to eliminate the matrix R_2 in (3.1.2) as follows. Applying it to the second row of R_1 and the first row of R_2, we eliminate the (1,2)-entry of R_2. Since both R_1 and R_2 are Toeplitz, the whole superdiagonal of R_2 can be annihilated by applying the same rotation to the ith row of R_1 and the $(i-1)$st row of R_2, for $i = 2, \ldots, n$. Therefore, the triangular decomposition (3.1.3) can be obtained by repeatedly applying the above procedure and eliminating one superdiagonal of R_2 at a time. Since both R_1 and R_2 are Toeplitz, we operate on vectors only and so the calculation of (3.1.3) requires $O(n^2)$ flops.

4. SYMMETRIC RANK-REVEALING FACTORIZATION

Suppose we have decomposed T into a product of simpler matrices, viz., (3.1.3). We now initiate a rank-revealing factorization. First, we find a normalized vector z that approximates the eigenvector corresponding to the smallest (in magnitude) eigenvalue. Call the eigenvalue $\sigma_{\min}$. Then

$$Tz \approx \sigma_{\min} z.$$

Next, we use the technique in [7] to find circular rotations (call their product $\bar{V}^T$) that transform z into the nth unit vector e_n:

$$\bar{V}^T z = e_n \quad (\text{hence} \;\; \bar{V} e_n = z).$$

Apply the transpose of these rotations from the right on R. However, when a circular rotation is applied to the ith and $(i+1)$st columns of R, it creates a nonzero $(i+1, i)$ subdiagonal entry in R. To restore its triangular structure, we apply a quadratic rotation from the left on R to annihilate the newly created nonzero element. So, we get

$$T = \bar{V}\bar{R}^T\bar{D}\bar{R}\bar{V}^T, \tag{4.1}$$

where $\bar{R}$ is upper triangular and $\bar{D}$ is a signature matrix. To check for rank-revelation, we compute the last column of $\bar{R}^T\bar{D}\bar{R}$; call this vector y. We get that

$$y \equiv \bar{R}^T\bar{D}\bar{R}e_n = \bar{V}^T T\bar{V}e_n = \bar{V}^T Tz \approx \sigma_{\min}\bar{V}^T z = \sigma_{\min} e_n, \tag{4.2}$$

and thus

$$\|y\|_2^2 \approx \sigma_{\min}^2. \tag{4.3}$$

Note that (2.2.5) reduces to (4.2) and (4.3) when $k = 1$. If the estimated smallest eigenvalue is less than the predetermined tolerance, we deflate T. This deflation procedure can be repeated until the estimated eigenvalue exceeds the tolerance. Consequently, we reveal the rank of T and obtain the desired VSV decomposition of (2.2.3):

$$T = V \begin{pmatrix} J & F \\ 0 & K \end{pmatrix}^T \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \begin{pmatrix} J & F \\ 0 & K \end{pmatrix} V^T, \tag{4.4}$$

so that equation (2.2.5) will be satisfied. The factorization (4.4), including the eigenvector estimation, costs $O(kn^2)$ flops.

5. EXAMPLES

We present three examples to show how our new method performs as well as the SVD approach. All examples come from [6]. Figure 0 depicts the original signal vector $x \in R^{130}$. The noise w is a random vector generated in MATLAB with normal distribution, zero mean, and unit variance. It is scaled so that

$$\frac{\|w\|_2}{\|Tx\|_2} = 0.001,$$

which is also chosen as the tolerance τ. We wrote a program in MATLAB and ran it on a SUN4/280S. The simple inverse power method [3] was used for estimating eigenvectors, and the TSVD regularization was realized using the MATLAB function *svd*.

In our examples, we compute the condition number of T, viz. $\kappa_2(T)$, and its numerical rank $n - k$ (cf. (1.3)). We will see that the examples are ill-conditioned (calling for regularization), and that our new technique always calculates the rank correctly. We also determine the relative error ϵ of the two solution vectors x_T and x_S, computed by the TSVD and our new symmetric decomposition, respectively. We define this relative error by

$$\epsilon = \frac{\|x_T - x_S\|_2}{\|x_T\|_2}.$$

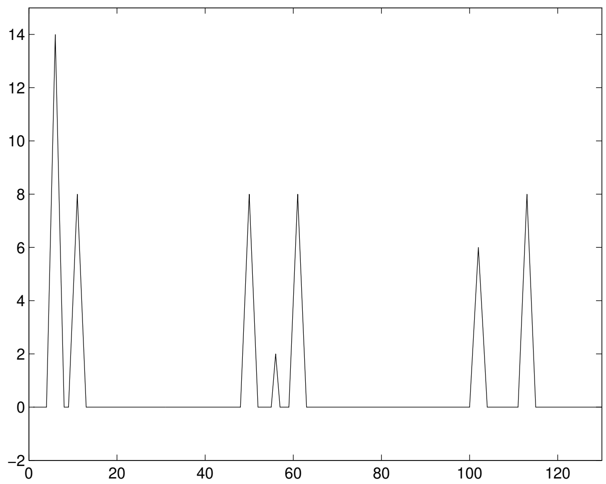

Figure 0. Original signal vector $x \in R^{130}$.

Example 1. The 130×130 matrix T is given by

$$t_m = \begin{cases} 1, & \text{if } m = 1; \\ \frac{1}{((m-1)/\omega)^2} \sin^2((m-1)/\omega), & \text{if } 2 \leq m \leq 8; \\ 0, & \text{otherwise;} \end{cases}$$

with $\omega = 4.0$. The matrix T is ill-conditioned in that $\kappa_2(T) \approx 1.2 \times 10^7$. The numerical rank detected by both methods is 128; hence $k = 2$ (a small rank deficiency). Results are shown in Figure 1; the relative error ϵ equals 2.7×10^{-4}.

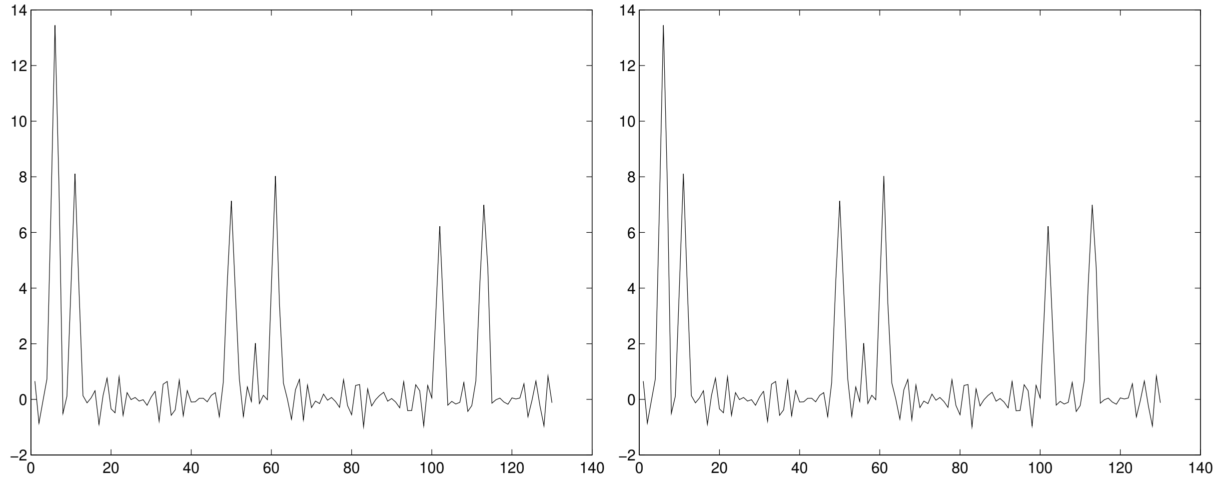

Figure 1. Results of Example 1. The graphs show x_T on the left and x_S on the right.

Example 2. The matrix T is generated by

$$t_m = \begin{cases} 2\omega, & \text{if } m = 1; \\ \frac{1}{\pi(m-1)} \sin(2\pi\omega(m-1)), & \text{if } 2 \le m \le 8; \\ 0, & \text{otherwise}; \end{cases}$$

with $\omega = 0.05$ and $\kappa_2(T) \approx 7.5 \times 10^6$. We calculated $k = 4$ from both methods. The relative error ϵ equals 7.2×10^{-3} and the solution vectors are shown in Figure 2.

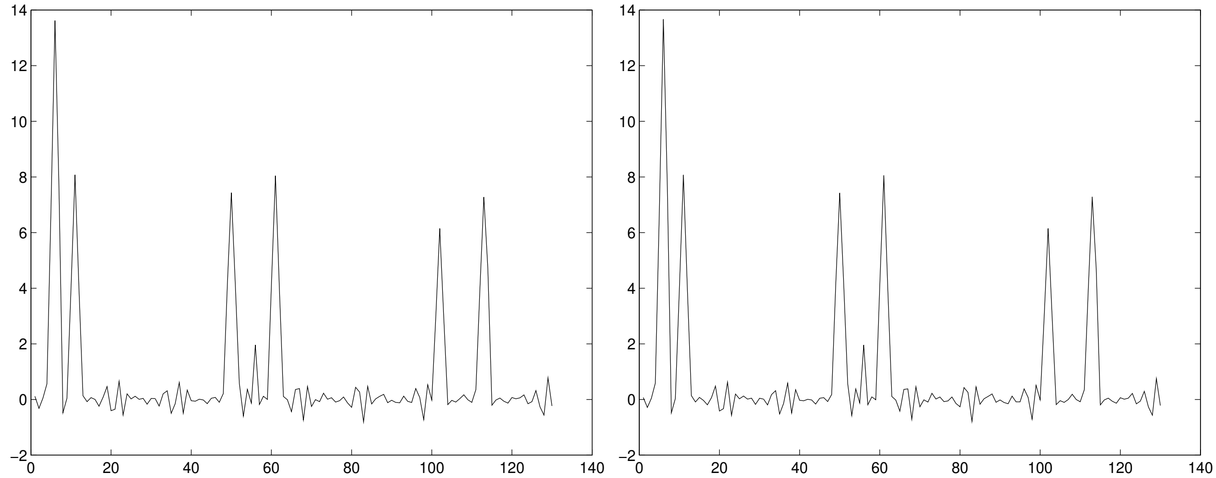

Figure 2. Results of Example 2. The graphs show x_T on the left and x_S on the right.

Example 3. We choose the 130×130 matrix T:

$$t_m = \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(m-1)^2}{2\sigma^2}}, & \text{if } 1 \le m \le 8; \\ 0, & \text{otherwise}; \end{cases}$$

with $\sigma = 2.0$ and $\kappa_2(T) \approx 1.7 \times 10^6$. Note that $k = 51$ with both methods. This example shows that our new approach still performs well when the rank deficiency is large; the error ϵ equals 3.1×10^{-3} (see Figure 3).

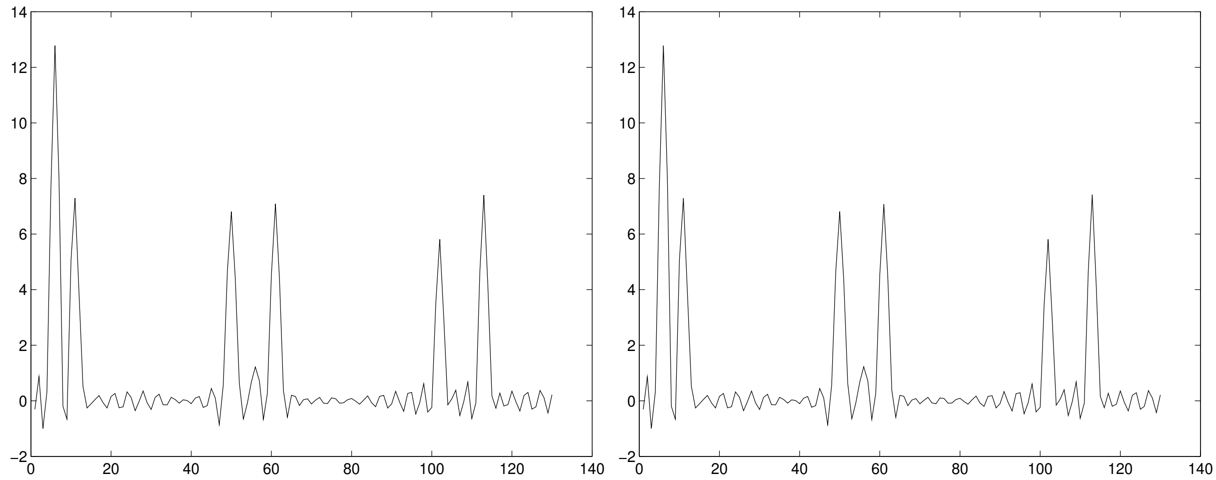

Figure 3. Results of Example 3. The graphs show x_T on the left and x_S on the right.

ACKNOWLEDGEMENTS

F.T. Luk was supported in part by the Office of Naval Research under contract N00014-93-1-0268, and S. Qiao by the Natural Sciences and Engineering Research Council of Canada under grant OGP0046301. The authors acknowledge valuable comments by David Vandevoorde.

REFERENCES

[1] H.C. Andrews and B.R. Hunt, *Digital Image Restoration,* Prentice-Hall, Englewood Cliffs, NJ, 1977.

[2] J. Chun, T. Kailath and H. Lev-Ari, "Fast parallel algorithms for QR and triangular factorization," *SIAM J. Sci. Stat. Comput.,* 8 (1987), No. 6, pp. 899-913.

[3] G.H. Golub and C.F. Van Loan, *Matrix Computations,* 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[4] P.C. Hansen, "The truncated SVD as a method for regularization," *BIT,* 27 (1987), pp. 534-553.

[5] F.T. Luk and D. Vandevoorde, "SVD-based analysis of image boundary distortion," in *SVD and Signal Processing, III: Algorithms, Analysis and Applications*, M. Moonen and B. De Moor, Eds., Elsevier, Amsterdam, pp. 305-313, 1995.

[6] J.G. Nagy, "Fast algorithms for the regularization of banded Toeplitz least squares problems," in Proc. SPIE 2296, *Advanced Signal Processing: Algorithms, Architectures, and Implementations V,* F.T. Luk, Ed., SPIE, Bellingham, WA, pp. 566-575, 1994.

[7] G.W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Proc.,* 40 (1992), pp. 1535-1541.