A Fast Singular Value Algorithm for Hankel Matrices

Franklin T. Luk and Sanzheng Qiao

Abstract. We present an $O(n^2 \log n)$ algorithm for finding all the singular values of an n-by-n complex Hankel matrix. We take advantage of complex symmetry and the Hankel structure. Our method is based on a modified Lanczos process and the Fast Fourier Transform.

1. Introduction

Structured matrices play an important role in signal processing. A common occurring structure is the Hankel form:

$$H = \begin{pmatrix} h_1 & h_2 & \cdots & h_{n-1} & h_n \\ h_2 & h_3 & \cdots & h_n & h_{n+1} \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-3} & h_{2n-2} \\ h_n & h_{n+1} & \cdots & h_{2n-2} & h_{2n-1} \end{pmatrix},$$

where all elements along the same anti-diagonal are identical. There is an extensive literature on inverting Hankel matrices or solving linear equations with a Hankel structure; for a good list of references, see Golub-Van Loan[**4**]. The work is more limited on efficient eigenvalue computation for Hankel matrices; some examples are Cybenko-Van Loan[**2**], Luk-Qiao[**6**] and Trench[**7**].

In this paper, we present an $O(n^2 \log n)$ algorithm for finding all the singular values of an n-by-n complex Hankel matrix H. Specifically, we present an algorithm for computing the Takagi decomposition (Horn-Johnson[**5**]):

$$H = Q\Sigma Q^{\mathrm{T}}, \tag{1}$$

where Q is unitary and Σ is diagonal with the singular values of H on its diagonal. An $O(n^3)$ algorithm for computing the Takagi decomposition of a general complex-symmetric matrix is given by Bunse-Gerstner and Gragg [**1**]. Their algorithm consists of two stages. First, a complex-symmetric matrix is reduced to a tridiagonal form using Householder transformations. Second, a complex-symmetric tridiagonal matrix is diagonalized by the QR method. They state [**1**, page 42]:

2000 *Mathematics Subject Classification.* Primary 15A23, 65F23.
Key words and phrases. Complex Hankel matrix, Takagi decomposition, Lanczos process.

> In many applications the symmetric matrix is a Hankel matrix and it is an open question whether there is a method for the symmetric SVD computation which can exploit this by far more special structure to reduce the number of operations and the storage requirement.

Nonetheless, it is remarked [**1**, page 51] that an n-by-n Hankel matrix H "can in principle be tridiagonalized by a Takagi-Lanczos process in $O(n^2 \log_2 n)$ operations, using FFT's to compute $H\mathbf{x}$." However, no further details are presented. We use the Lanczos and FFT procedures in this paper, before we became aware of the work by Bunse-Gerstner and Gragg [**1**].

Our paper is organized as follows. A Lanczos tridiagonalization of a Hankel matrix is described in Section 2, and a two-by-two Takagi decomposition in Section 3. A QR method for the diagonalization of a tridiagonal complex-symmetric matrix is given in Section 4, followed by an overall algorithm and an illustrative numerical example in Section 5.

Notations. We use the "bar" symbol to denote a complex conjugate; for example, $\bar{M}$, $\bar{\mathbf{v}}$ and $\bar{\alpha}$ denote the complex conjugates of a matrix M, a vector $\mathbf{v}$ and a scalar α, respectively.

2. Lanczos Tridiagonalization

As in the standard SVD computation, we first reduce the Hankel matrix to a simpler form, specifically a complex-symmetric tridiagonal matrix.

Consider a Lanczos-like algorithm to reduce H to an upper Hessenberg form:

$$H\bar{Q} = QK, \tag{2}$$

where the matrix Q is unitary and the matrix K is upper Hessenberg. Let

$$Q = (\ \mathbf{q}_1,\ \mathbf{q}_2, \ldots,\ \mathbf{q}_n\ )$$

and

$$K = (\kappa_{ij}).$$

Writing out equation (2) in column form, we obtain a recurrence formula to compute Q and K:

$$\kappa_{l+1,l}\mathbf{q}_{l+1} = H\bar{\mathbf{q}}_l - \kappa_{l,l}\mathbf{q}_l - \kappa_{l-1,l}\mathbf{q}_{l-1} - \cdots - \kappa_{1,l}\mathbf{q}_1. \tag{3}$$

Since

$$K = Q^H H \bar{Q},$$

the matrix K is complex-symmetric. From symmetry, we deduce that K is tridiagonal:

$$K = \begin{pmatrix} \alpha_1 & \beta_1 & & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & & \\ & \beta_2 & \alpha_3 & \beta_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & & & & \beta_{n-1} & \alpha_n \end{pmatrix}. \tag{4}$$

The relation (3) thus reduces to a three term recursion:

$$\beta_l \mathbf{q}_{l+1} = H\bar{\mathbf{q}}_l - \alpha_l \mathbf{q}_l - \beta_{l-1}\mathbf{q}_{l-1}. \tag{5}$$

As $\mathbf{q}_i^{\mathrm{H}}\mathbf{q}_j = \delta_{ij}$, we get from (5) that $\alpha_l = \mathbf{q}_l^{\mathrm{H}} H \bar{\mathbf{q}}_l$. Let

$$\mathbf{r}_l = H\bar{\mathbf{q}}_l - \alpha_l \mathbf{q}_l - \beta_{l-1}\mathbf{q}_{l-1}.$$

Then (5) shows that β_l is the 2-norm of $\mathbf{r}_l$ and $\mathbf{q}_{l+1}$ is the normalized $\mathbf{r}_l$, i.e.,

$$\beta_l = \sqrt{\mathbf{r}_l^{\mathrm{H}}\mathbf{r}_l}$$

and

$$\mathbf{q}_{l+1} = \mathbf{r}_l/\beta_l.$$

The triangularization procedure is summarized in the following algorithm.

ALGORITHM 1 (Lanczos Tridiagonalization). *Given an n-by-n Hankel matrix H, this algorithm computes a unitary matrix Q such that $H = QKQ^{\mathrm{T}}$, where K is complex-symmetric and tridiagonal as shown in (4).*

Initialize $\mathbf{q}_1$ such that $\|\mathbf{q}_1\|_2 = 1$;
Set $\mathbf{r}_0 = \mathbf{q}_1$; $\beta_0 = 1$; $\mathbf{q}_0 = \mathbf{0}$; $l = 0$;
while ($\beta_l \neq 0$)
    $\mathbf{q}_{l+1} = \mathbf{r}_l/\beta_l$;
    $l \leftarrow l + 1$;
    $\alpha_l = \mathbf{q}_l^{\mathrm{H}} H\bar{\mathbf{q}}_l$;
    $\mathbf{r}_l = H\bar{\mathbf{q}}_l - \alpha_l\mathbf{q}_l - \beta_{l-1}\mathbf{q}_{l-1}$;
    $\beta_l = \|\mathbf{r}_l\|_2$;
end. □

If all $\beta_l \neq 0$, then Algorithm 1 runs until $l = n$. The dominant cost is the Hankel matrix-vector product $H\bar{\mathbf{q}}_l$, for $l = 1, 2, \ldots, n$. Using an $O(n \log n)$ Hankel matrix-vector multiplication scheme (cf. Luk-Qiao[**6**]), we obtain an $O(n^2 \log n)$ tridiagonalization algorithm.

3. Two-by-Two Takagi Decomposition

In this section, we discuss the Takagi decomposition (1) of a 2-by-2 complex-symmetric matrix.

Consider

$$A = \begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix}. \tag{6}$$

We look for a unitary matrix Q such that

$$Q^{\mathrm{H}} A \bar{Q} = \Sigma, \tag{7}$$

where

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

and both σ_1 and σ_2 are nonnegative. Define

$$\operatorname{sign}(x) = \begin{cases} x/|x| & \text{if } x \neq 0, \\ 1 & \text{if } x = 0. \end{cases}$$

If $\beta = 0$, then we pick

$$Q = \begin{pmatrix} \sqrt{\operatorname{sign}(\alpha)} & 0 \\ 0 & \sqrt{\operatorname{sign}(\gamma)} \end{pmatrix}, \tag{8}$$

which gives

$$Q^{\mathrm{H}} A \bar{Q} = \begin{pmatrix} |\alpha| & 0 \\ 0 & |\gamma| \end{pmatrix}.$$

So we assume $\beta \neq 0$ from here on. The product $A\bar{A}$ $(= AA^{\mathrm{H}})$ is Hermitian and nonnegative definite. Indeed,

$$A\bar{A} = \begin{pmatrix} |\alpha|^2 + |\beta|^2 & \alpha\bar{\beta} + \beta\bar{\gamma} \\ \bar{\alpha}\beta + \bar{\beta}\gamma & |\beta|^2 + |\gamma|^2 \end{pmatrix}.$$

We can find an eigenvalue decomposition:

$$V^{\mathrm{H}}(A\bar{A})V = D^2, \tag{9}$$

where the matrix V is unitary and the matrix D nonnegative diagonal. Let

$$V = (\mathbf{v}_1, \mathbf{v}_2)$$

and

$$D = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}.$$

The Takagi decomposition implies that $A\bar{Q} = Q\Sigma$. So we look for a normalized vector $\mathbf{q}$ such that

$$A\bar{\mathbf{q}} = \sigma_i \mathbf{q}, \quad \sigma_i \geq 0, \tag{10}$$

from which we get

$$A\bar{A}\mathbf{q} = \sigma_i A\bar{\mathbf{q}} = \sigma_i^2 \mathbf{q}.$$

Hence $\mathbf{q}$ is an eigenvector of $A\bar{A}$ and σ_i^2 is the corresponding eigenvalue. Thus, σ_i^2 equals either d_1^2 or d_2^2. We have two cases depending on whether the eigenvalues d_1^2 and d_2^2 are distinct or identical.

First, we assume that $d_1 \neq d_2$. Since the eigenvalues of $A\bar{A}$ are distinct, the eigenvectors are uniquely defined up to a scalar. Thus, $\mathbf{q}$ is a scalar multiple of either $\mathbf{v}_1$ or $\mathbf{v}_2$ and σ_i is either d_1 or d_2, respectively. Let

$$A\bar{\mathbf{v}}_1 = \xi d_1 \mathbf{v}_1 \tag{11}$$

for some scalar ξ such that $|\xi| = 1$. We define

$$\mathbf{q} \equiv \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \sqrt{\operatorname{sign}(\xi)}\, \mathbf{v}_1. \tag{12}$$

Then

$$A\bar{\mathbf{q}} = \sqrt{\operatorname{sign}(\bar{\xi})} \cdot A\bar{\mathbf{v}}_1 = \sqrt{\operatorname{sign}(\bar{\xi})} \cdot \xi d_1 \mathbf{v}_1 = |\xi|\, d_1 \mathbf{q} = d_1 \mathbf{q},$$

as desired. We can get ξ from (11):

$$\xi = \operatorname{sign}(\mathbf{v}_1^{\mathrm{H}} A \bar{\mathbf{v}}_1). \tag{13}$$

Second, we assume that the eigenvalues are identical, i.e., $d_1 = d_2$. Then $A\bar{A} = d_1^2 I$ and any vector is an eigenvector. Pick $\mathbf{v}_1 = \mathbf{e}_1$. Then $A\bar{\mathbf{v}}_1 \neq \eta \mathbf{v}_1$ for any scalar η, since $\beta \neq 0$. We propose

$$\mathbf{u} = A\bar{\mathbf{e}}_1 + d_1\, \mathbf{e}_1 = \begin{pmatrix} \alpha + d_1 \\ \beta \end{pmatrix}.$$

Then

$$A\bar{\mathbf{u}} = A\bar{A}\mathbf{e}_1 + d_1 A\bar{\mathbf{e}}_1 = d_1^2 \mathbf{e}_1 + d_1 A\bar{\mathbf{e}}_1 = d_1 \mathbf{u}.$$

We choose $\mathbf{q}$ as a normalized $\mathbf{u}$ with $q_1 \geq 0$. Let

$$\mathbf{q} \equiv \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \frac{1}{\sqrt{|\alpha + d_1|^2 + |\beta|^2}} \begin{pmatrix} |\alpha + d_1| \\ \beta(\bar{\alpha} + \bar{d}_1)/|\alpha + d_1| \end{pmatrix}. \tag{14}$$

Given $\mathbf{q}$ in either (12) or (14), we can construct a unitary matrix Q by

$$Q = \begin{pmatrix} q_1 & -\bar{q}_2 \\ q_2 & \bar{q}_1 \end{pmatrix}. \tag{15}$$

Thus, the product

$$Q^{\mathrm{H}} A \bar{Q} = \begin{pmatrix} d_1 & \times \\ 0 & \times \end{pmatrix}$$

is upper triangular. As the matrix $Q^{\mathrm{H}} A \bar{Q}$ is symmetric, it is therefore diagonal. In fact, we can readily show that

$$Q^{\mathrm{H}} A \bar{Q} = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix},$$

assuming that the second column of Q has been scaled so that $d_2 \geq 0$.

ALGORITHM 2 (2-by-2 Takagi Decomposition). *Given a 2-by-2 complex symmetric A in (6), this algorithm computes a 2-by-2 unitary matrix Q so that (7) is satisfied.*

If $\beta = 0$ then pick Q as in (8)
else
  Find an eigenvalue decomposition of $A\bar{A}$: $V^{\mathrm{H}} A\bar{A} V = D^2$;
  If $d_1 \neq d_2$ then set $\mathbf{q}$ using (12) (ξ from (13)) else set $\mathbf{q}$ using (14);
  Construct unitary matrix Q as in (15)
end. □

4. Diagonalization

This section concerns the Takagi decomposition of an n-by-n tridiagonal matrix K of (4):

$$K = Q \Sigma Q^{\mathrm{T}},$$

where Q is unitary and Σ is nonnegative diagonal. We apply an implicit QR algorithm; that is, instead of an explicit formation of $K^{\mathrm{H}} K$, we apply Householder transformations directly to K.

Consider the trailing 3-by-3 submatrix Λ of of $K^{\mathrm{H}} K$. It is given by

$$\Lambda = \begin{pmatrix} \lambda_{11} & \bar{\alpha}_{n-2}\beta_{n-2} + \alpha_{n-1}\bar{\beta}_{n-2} & \bar{\beta}_{n-2}\beta_{n-1} \\ \alpha_{n-2}\bar{\beta}_{n-2} + \bar{\alpha}_{n-1}\beta_{n-2} & \lambda_{22} & \bar{\alpha}_{n-1}\beta_{n-1} + \alpha_n\bar{\beta}_{n-1} \\ \bar{\beta}_{n-1}\beta_{n-2} & \alpha_{n-1}\bar{\beta}_{n-1} + \bar{\alpha}_n\beta_{n-1} & \lambda_{33} \end{pmatrix}, \tag{16}$$

where

$$\begin{aligned} \lambda_{11} &= |\beta_{n-3}|^2 + |\alpha_{n-2}|^2 + |\beta_{n-2}|^2, \\ \lambda_{22} &= |\beta_{n-2}|^2 + |\alpha_{n-1}|^2 + |\beta_{n-1}|^2, \\ \lambda_{33} &= |\beta_{n-1}|^2 + |\alpha_n|^2. \end{aligned} \tag{17}$$

Let λ denote the eigenvalue of Λ of that is the closest to λ_{33}, and let J_1 be the Householder matrix such that

$$J_1^T \mathbf{x} = (\times, 0, 0)^{\mathrm{T}},$$

where

$$\mathbf{x} = \begin{pmatrix} |\alpha_1|^2 + |\beta_1|^2 - \lambda \\ \alpha_1\bar{\beta}_1 + \bar{\alpha}_2\beta_1 \\ \beta_1\bar{\beta}_2 \end{pmatrix}. \tag{18}$$

We note that the vector $\mathbf{x}$ consists of the top three elements from the first column of $K^{\mathrm{H}}K - \lambda I$. Let us apply

$$\widetilde{J}_1 = \begin{pmatrix} J_1^T & 0 \\ 0 & 1 \end{pmatrix}$$

to a block in K directly:

$$\begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 & \delta_1 \\ \beta_1 & \alpha_2 & \beta_2 & \gamma_2 \\ \gamma_1 & \beta_2 & \alpha_3 & \beta_3 \\ \delta_1 & \gamma_2 & \beta_3 & \alpha_4 \end{pmatrix} \leftarrow \widetilde{J}_1^{\mathrm{T}} \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \beta_2 & \\ & \beta_2 & \alpha_3 & \beta_3 \\ & & \beta_3 & \alpha_4 \end{pmatrix} \widetilde{J}_1. \tag{19}$$

We follow with Householder transformations J_2, J_3, ..., J_{n-2} to restore the tridiagonal structure of K, while maintaining symmetry at the same time. To illustrate, we determine a Householder matrix J_k such that

$$J_k^{\mathrm{T}}(\beta_{k-1}, \gamma_{k-1}, \delta_{k-1})^{\mathrm{T}} = (\times, 0, 0)^{\mathrm{T}}.$$

Let

$$\widetilde{J}_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & J_k^{\mathrm{T}} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Then

$$\begin{pmatrix} \alpha_{k-1} & \beta_{k-1} & & & \\ \beta_{k-1} & \alpha_k & \beta_k & \gamma_k & \delta_k \\ & \beta_k & \alpha_{k+1} & \beta_{k+1} & \gamma_{k+1} \\ & \gamma_k & \beta_{k+1} & \alpha_{k+2} & \beta_{k+2} \\ & \delta_k & \gamma_{k+1} & \beta_{k+2} & \alpha_{k+3} \end{pmatrix} \leftarrow \tag{20}$$

$$\widetilde{J}_k^{\mathrm{T}} \begin{pmatrix} \alpha_{k-1} & \beta_{k-1} & \gamma_{k-1} & \delta_{k-1} & \\ \beta_{k-1} & \alpha_k & \beta_k & \gamma_k & \\ \gamma_{k-1} & \beta_k & \alpha_{k+1} & \beta_{k+1} & \\ \delta_{k-1} & \gamma_k & \beta_{k+1} & \alpha_{k+2} & \beta_{k+2} \\ & & & \beta_{k+2} & \alpha_{k+3} \end{pmatrix} \widetilde{J}_k.$$

Consider the new matrix $K^{(k)} \leftarrow \widehat{J}_k^{\mathrm{T}} \cdots \widehat{J}_2^{\mathrm{T}} \widehat{J}_1^{\mathrm{T}} K \widehat{J}_1 \widehat{J}_2 \cdots \widehat{J}_k$, where $\widehat{J}_i$ denotes the appropriate $n \times n$ Householder matrix that contains the lower-dimension $\widetilde{J}_i$ as a submatrix. The resultant matrix is symmetric and the bulge is chased down by one row and one column. Eventually, the bulge is chased out of the matrix and the final new matrix

$$K \equiv K^{(n-1)} \leftarrow \widehat{J}_{n-1}^{\mathrm{T}} \cdots \widehat{J}_2^{\mathrm{T}} \widehat{J}_1^{\mathrm{T}} K \widehat{J}_1 \widehat{J}_2 \cdots \widehat{J}_{n-1} \tag{21}$$

is symmetric and tridiagonal. As in the standard implicit QR method, the new K is closer to diagonal. Thus, we have derived an algorithm for one step of a complex-symmetric SVD.

ALGORITHM 3 (One Step of Complex-Symmetric SVD). *Given diagonal vector*

$$\mathbf{a} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^{\mathrm{T}}$$

and subdiagonal vector

$$\mathbf{b} = (\beta_1, \beta_2, \ldots, \beta_{n-1})^{\mathrm{T}}$$

of a tridiagonal matrix K of (4), this algorithm overwrites $\mathbf{a}$ *and* $\mathbf{b}$ *so that the new matrix K formed by the new* $\mathbf{a}$ *and* $\mathbf{b}$ *is the same as the matrix obtained by applying one step of an implicit QR algorithm.*

If $n = 2$, apply Algorithm 2 and return;
Compute λ, the eigenvalue of Λ of (16) that is closest to λ_{33} of (17);
for $k = 1 : n-2$
    if $k = 1$ then set $\mathbf{x}$ using (18) else set $\mathbf{x} = (\beta_{k-1},\ \gamma_{k-1},\ \delta_{k-1})^{\mathrm{T}}$;
    Determine a Householder matrix J_k such that $J_k^{\mathrm{T}}\mathbf{x} = (\times, 0, 0)^{\mathrm{T}}$;
    if $k = 1$ then update block as in (19) else update block as in (20);
    end
end
Determine a Householder matrix J_{n-1} s.t. $J_{n-1}^{\mathrm{T}}(\beta_{n-2}, \gamma_{n-2})^{\mathrm{T}} = (\times, 0)^{\mathrm{T}}$;
Update the last block:

$$\begin{pmatrix} \alpha_{n-2} & \beta_{n-2} & 0 \\ \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & \beta_{n-1} & \alpha_n \end{pmatrix} \leftarrow$$

$$\begin{pmatrix} 1 & 0 \\ 0 & J_{n-1}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \alpha_{n-2} & \beta_{n-2} & \gamma_{n-2} \\ \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ \gamma_{n-2} & \beta_{n-1} & \alpha_n \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & J_{n-1} \end{pmatrix}. \quad \Box$$

The one-step algorithm requires $O(n)$ flops. Accumulating J_i requires $O(n^2)$ flops. The method is used in the singular value computation. Let q be the largest integer such that

$$\mathbf{b}_{n-q:n} = \mathbf{0},$$

i.e., the subvector $\mathbf{b}_{n-q:n}$ is the null vector. (Initially, we have $q = 0$ and $\beta_n = 0$.) Also, let p denote the smallest integer such that the subvector $\mathbf{b}_{p+1:n-q-1}$ has no zero entries. Then the principal submatrix

$$B = \begin{pmatrix} \alpha_{p+1} & \beta_{p+1} & & & 0 \\ \beta_{p+1} & \alpha_{p+2} & \beta_{p+2} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-q-1} \\ 0 & & & \beta_{n-q-1} & \alpha_{n-q} \end{pmatrix}$$

has no zeros on its subdiagonal. We apply Algorithm 3 to B. When some β_i becomes sufficiently small:

$$|\beta_i| \le c(|\alpha_i| + |\alpha_{i+1}|)u, \tag{22}$$

where c denotes a small constant and u the unit roundoff, then we set β_i to zero and update p and q. When q reaches $n-1$, K becomes diagonal with the singular values on the diagonal.

ALGORITHM 4 (Complex-symmetric SVD). *Given the diagonal* $\mathbf{a}$ *and subdiagonal* $\mathbf{b}$ *of the tridiagonal matrix* K *of (4), this algorithm computes the Takagi decomposition* $K = Q\Sigma Q^{\mathrm{T}}$. *The diagonal* $\mathbf{a}$ *is overwritten by the singular values.*

```
Initialize q = 0, β_n = 0, and Q = I;
while q < n − 1
    Set all β_i satisfying (22) to zero;
    Update q so that b_{n−q:n} = 0 and β_{n−q−1} ≠ 0;
    Find the smallest p so that b_{p+1:n−q−1} has no zero entries;
    If q < n − 1
        Apply Algorithm 3 to the complex-symmetric and
        tridiagonal matrix whose diagonal and subdiagonal
        are a_{p+1:n−q} and b_{p+1:n−q−1}, respectively;
        Update Q;
    end
end.    □
```

As stated before, Algorithm 3 requires $O(n)$ flops without accumulating the Householder transformations. Thus, Algorithm 4 uses $O(n^2)$ flops without explicitly forming Q.

5. Overall Procedure

We conclude our paper with an overall singular value procedure and an illustrative numerical example.

ALGORITHM 5 (Fast Hankel Singular Value Algorithm). *Given a complex Hankel matrix* H, *this algorithm computes all its singular values.*

1. Apply Algorithm 1 to H to obtain a symmetric and tridiagonal K;
2. Apply Algorithm 4 to calculate the singular values of K. □

The major cost of this algorithm is the tridiagonalization procedure and the dominant cost of the Lanczos tridiagonalization is matrix-vector multiplication. The $O(n \log n)$ Hankel matrix-vector multiplication scheme is faster than general $O(n^2)$ matrix-vector multiplication when $n \geq 16$ (cf. Luk-Qiao [**6**]). We expect our proposed Hankel SVD algorithm to be faster than a general SVD algorithm for a very small value of n.

We know that, in a straightforward implementation of Lanczos procedure, the orthogonality of the vectors $\mathbf{q}_l$ in Algorithm 1 deteriorates as the size of H increases. Reorthogonalization is necessary for a practical Lanczos method. Efficient and practical reorthogonalization techniques are available, see [**3**, §7.5] and references there. They achieve the orthogonality of $\mathbf{q}_l$ nearly as good as complete reorthogonalization with just a little extra work.

Example. Suppose that the first column and the last row of H are respectively

$$\begin{pmatrix} 0.9501 + 0.7621i \\ 0.2311 + 0.4565i \\ 0.6068 + 0.0185i \\ 0.4860 + 0.8214i \\ 0.8913 + 0.4447i \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0.8913 + 0.4447i \\ 0.7919 + 0.9355i \\ 0.9218 + 0.9169i \\ 0.7382 + 0.4103i \\ 0.1763 + 0.8937i \end{pmatrix}.$$

After tridiagonalization, the diagonal and subdiagonal of K are respectively

$$\begin{pmatrix} 3.4438 + 3.0893i \\ 0.1558 + 0.1970i \\ 0.1729 + 0.0537i \\ 0.3771 + 0.0265i \\ -0.7437 + 0.4832i \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0.5400 \\ 0.6584 \\ 0.5859 \\ 0.4940 \end{pmatrix}.$$

The following table presents the four subdiagonal elements of K during the execution of Algorithm 4.

Iter.	β_1	β_2	β_3	β_4
1	$-0.017 + 0.000i$	$0.491 - 0.087i$	$-0.531 + 0.115i$	$-0.233 - 0.041i$
2	10^{-4}	$0.389 - 0.071i$	$0.215 - 0.050i$	10^{-3}
3	10^{-5}	$0.318 - 0.058i$	$-0.072 + 0.017i$	10^{-8}
4	10^{-6}	$0.253 - 0.046i$	$0.024 - 0.006i$	converged
5	10^{-8}	$0.209 - 0.038i$	converged	
6	10^{-9}	10^{-15}		
7	10^{-11}	converged		
8	converged			

The computed singular values are $\{\ 4.6899,\ 1.1819,\ 1.0673,\ 0.62109,\ 0.37028\ \}$. Assuming that the MATLAB function svd( ) is fully accurate, we find the errors in the singular values computed by Algorithm 5 to be 10^{-15}.

Acknowledgement. The authors thank Professor Angelika Bunse-Gerstner for her valuable comment.

References

[1] A. Bunse-Gerstner and W. B. Gragg, "Singular value decompositions of complex symmetric matrices," *J. Comput. Appl. Math.*, 21, 1988, pp. 41–54.

[2] G. Cybenko and C. F. Van Loan, "Computing the minimum eigenvalue of a symmetric positive definite Toeplitz matrix," *SIAM J. Sci. Statist. Comput.*, 7, 1986, pp. 123–131.

[3] J. W. Demmel, *Applied Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, 1997.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd Ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[5] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[6] F. T. Luk and S. Qiao, "A fast eigenvalue algorithm for Hankel matrices," *Lin. Alg. Applics.*, 316, 2000, pp. 171–182.

[7] W. F Trench, "Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices," *SIAM J. Matrix Anal. Appl.*, Vol. 10, No. 2, 1989, pp. 135–146.

DEPT OF COMPUTER SCIENCE, RENSSELAER POLYTECHNIC INSTITUTE, TROY, NY 12180, USA
E-mail address: `luk@cs.rpi.edu`

DEPT OF COMPUTING AND SOFTWARE, MCMASTER UNIV., HAMILTON, ONT L8S 4L7, CANADA
E-mail address: `qiao@mcmaster.ca`