Analysis of a Fast Hankel Eigenvalue Algorithm

Franklin T. Luk[a] and Sanzheng Qiao[b]

[a]Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180 USA

[b]Department of Computing and Software
McMaster University
Hamilton, Ontario L8S 4L7 Canada

ABSTRACT

This paper analyzes the important steps of an $O(n^2 \log n)$ algorithm for finding the eigenvalues of a complex Hankel matrix. The three key steps are a Lanczos-type tridiagonalization algorithm, a fast FFT-based Hankel matrix-vector product procedure, and a QR eigenvalue method based on complex-orthogonal transformations. In this paper, we present an error analysis of the three steps, as well as results from numerical experiments.

Keywords: Hankel matrix, eigenvalue decomposition, Lanczos tridiagonalization, Hankel matrix-vector multiplication, complex-orthogonal transformations, error analysis.

1. INTRODUCTION

The eigenvalue decomposition of a structured matrix has important applications in signal processing. In this paper, we consider a complex Hankel matrix $H \in \mathcal{C}^{n \times n}$:

$$H \equiv \begin{pmatrix} h_1 & h_2 & \dots & h_{n-1} & h_n \\ h_2 & h_3 & \dots & h_n & h_{n+1} \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ h_{n-1} & h_n & \dots & h_{2n-3} & h_{2n-2} \\ h_n & h_{n+1} & \dots & h_{2n-2} & h_{2n-1} \end{pmatrix}. \tag{1}$$

The authors[4] proposed a fast algorithm for finding the eigenvalues of H. The key step is a fast Lanczos tridiagonalization algorithm which employs a fast Hankel matrix-vector multiplication based on the Fast Fourier transform (FFT). Then the algorithm performs a QR-like procedure using the complex-orthogonal transformations in the diagonalization to find the eigenvalues. In this paper, we present an error analysis and discuss the stability properties of the key steps of the algorithm.

This paper is organized as follows. In Section 2, we discuss eigenvalue computation using complex-orthogonal transformations, and describe the overall eigenvalue procedure and three key steps. In Section 3, we introduce notations and error analysis of some basic complex arithmetic operations. We analyze the orthogonality of the Lanczos vectors in Section 4 and the accuracy of the Hankel matrix-vector multiplication in Section 5. In Section 6, we derive the condition number of a complex-orthogonal plane rotation and we provide a backward error analysis of the application of this transformation to a vector. Numerical results are given in Section 7.

Supported by the Natural Sciences and Engineering Research Council of Canada under grant OGP0046301

2. COMPLEX-ORTHOGONAL TRANSFORMATIONS

An eigenvalue decomposition of a nondefective matrix H is given by

$$H = XDX^{-1}. \tag{2}$$

We will pick the matrix X to be *complex-orthogonal*; that is, $XX^{\mathrm{T}} = I$. So, $H = XDX^{\mathrm{T}}$. We apply a special Lanczos tridiagonalization to the Hankel matrix (assuming that the Lanczos process does not prematurely terminate):

$$H = QJQ^{\mathrm{T}}, \tag{3}$$

where Q is complex-orthogonal and J is complex-symmetric tridiagonal:

$$J \equiv \begin{pmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ 0 & & & \beta_{n-1} & \alpha_n \end{pmatrix}. \tag{4}$$

Let $Q \equiv (\mathbf{q}_1 \quad \mathbf{q}_2 \quad \dots \quad \mathbf{q}_n)$. Since $Q^{\mathrm{T}}Q = I$, we see that

$$\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_j = \delta_{ij}. \tag{5}$$

We call the property defined by (5) as complex-orthogonality (*c-orthogonality* for short).

The dominant cost of the Lanczos method is matrix-vector multiplication. We propose an $O(n \log n)$ FFT-based multiplication scheme so that we can tridiagonalize a Hankel matrix in $O(n^2 \log n)$ operations. Given $\mathbf{w} \in \mathcal{C}^n$ we want to compute the product $\mathbf{p} = H\mathbf{w}$. Define $\widehat{\mathbf{c}} \in \mathcal{C}^{2n-1}$ by

$$\widehat{\mathbf{c}} = (h_n \quad h_{n+1} \quad \dots \quad h_{2n-1} \quad h_1 \quad h_2 \quad \dots \quad h_{n-1})^{\mathrm{T}}. \tag{6}$$

Let $\mathbf{w} \equiv (w_1 \quad w_2 \quad \dots \quad w_n)^{\mathrm{T}}$. Define $\widehat{\mathbf{w}} \in \mathcal{C}^{2n-1}$ by

$$\widehat{\mathbf{w}} = (w_n \quad w_{n-1} \quad \dots \quad w_1 \quad 0 \quad \dots \quad 0)^{\mathrm{T}}. \tag{7}$$

Let $\mathrm{fft}(\mathbf{v})$ denote one-dimensional FFT of a vector $\mathbf{v}$, $\mathrm{ifft}(\mathbf{v})$ one-dimensional inverse FFT of $\mathbf{v}$, and ".*" component-wise multiplication of two vectors. The desired product is given by the first n elements of the vector $\mathrm{ifft}(\mathrm{fft}(\widehat{\mathbf{c}}).*\mathrm{fft}(\widehat{\mathbf{w}}))$. The authors[4] showed that Algorithm 2 requires $30n\log(n) + O(n)$ flops (real floating-point operations).

To maintain symmetry and tridiagonality of J, we apply complex-orthogonal rotations in the QR iterations:

$$J = WDW^{\mathrm{T}},$$

where W is complex-orthogonal and D is diagonal. Thus, we get (2) with $X = QW$. The workhorse of our QR-like iteration[3] is a complex-orthogonal "plane-rotation" $P \in \mathcal{C}^{n\times n}$. That is, P is an identity matrix except for four strategic positions formed by the intersection of the ith and jth rows and columns:

$$G = \begin{pmatrix} p_{ii} & p_{ij} \\ p_{ji} & p_{jj} \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \tag{8}$$

where $c^2 + s^2 = 1$. The matrix G is constructed to annihilate the second component of a 2-vector $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$.

We present our computational schemes in the following.

Overall Procedure (Hankel Eigenvalue Computation). *Given H, compute all its eigenvalues.*

1. Apply a Lanczos procedure (Algorithm 1) to reduce H to a tridiagonal form J. Use Algorithm 2 to compute the Hankel matrix-vector products.
2. Apply a QR-type procedure (Algorithm 3) to find the eigenvalues of J. □

ALGORITHM 1 (LANCZOS TRIDIAGONALIZATION). *Given H, this algorithm computes a tridiagonal matrix J.*

Initialize $\mathbf{q}_1$ such that $\mathbf{q}_1^{\mathrm{T}}\mathbf{q}_1 = 1$;
$\mathbf{v}_1 = H\mathbf{q}_1$;
for $j = 1 : n$
  $\alpha_j = \mathbf{q}_j^{\mathrm{T}}\mathbf{v}_j$;
  $\mathbf{r}_j = \mathbf{v}_j - \alpha_j\mathbf{q}_j$;
  if $j < n$
    $\beta_j = \sqrt{\mathbf{r}_j^{\mathrm{T}}\mathbf{r}_j}$;
    if $\beta_j = 0$, break; end;
    $\mathbf{q}_{j+1} = \mathbf{r}_j/\beta_j$;
    $\mathbf{v}_{j+1} = H\mathbf{q}_{j+1} - \beta_j\mathbf{q}_j$;
  end
end □

ALGORITHM 2 (FAST HANKEL MATRIX-VECTOR MULTIPLICATION). *Given $\mathbf{w} \in \mathcal{C}^n$, compute $\mathbf{p} = H\mathbf{w}$.*
Construct the vectors $\widehat{\mathbf{c}}$ and $\widehat{\mathbf{w}}$ as in (6) and (7). Calculate $\mathbf{y} \in \mathcal{C}^{2n-1}$ by

$$\mathbf{y} = \mathrm{ifft}(\mathrm{fft}(\widehat{\mathbf{c}}). * \mathrm{fft}(\widehat{\mathbf{w}})).$$

The desired $\mathbf{p} \in \mathcal{C}^n$ is given by the first n elements of $\mathbf{y}$: $\mathbf{p} = (\, y_1 \quad y_2 \quad \ldots \quad y_{n-1} \quad y_n \,)^{\mathrm{T}}$. □

ALGORITHM 3 (COMPLEX-SYMMETRIC QR STEP). *Given $J \in \mathcal{C}^{m\times m}$, this algorithm implements one step of the QR method with the Wilkinson shift.*

Initialize $Q = I$;
Find the eigenvalue μ of $J_{m-1:m,m-1:m}$ that is closer to J_{mm};
Set $x_1 = J_{11} - \mu$; $x_2 = J_{21}$;
for $k = 1 : m-1$
  Find a complex-orthogonal matrix P_k^{T} to annihilate x_2 using x_1;
  $J = P_k^{\mathrm{T}} J P_k$;
  $Q = QP_k$;
  if $k < m-1$
    $x_1 = J_{k+1,k}$; $x_2 = J_{k+2,k}$;
  end if
end for. □

3. ERROR ANALYSIS FUNDAMENTALS

Let u denote the unit roundoff. Since the objective of this paper is to analyze the stability of our algorithm[4] and not to derive precise error bounds, we carry out first-order error analysis. Sometimes, we ignore moderate constant coefficients. The following results for the basic complex arithmetic operations are from Higham[2]:

$$\begin{aligned} fl(x \pm y) &= (x \pm y)(1+\delta), & |\delta| &\le u, & (9)\\ fl(xy) &= xy(1+\delta), & |\delta| &\le \sqrt{2}\,\gamma_2, & (10)\\ fl(x/y) &= (x/y)(1+\delta), & |\delta| &\le \sqrt{2}\,\gamma_4, & (11)\end{aligned}$$

Using (10) and, similar to the real case,[2] repeatedly applying (9), we can show that for two n-vectors $\mathbf{x}$ and $\mathbf{y}$

$$fl(\mathbf{x}^{\mathrm{T}}\mathbf{y}) = \mathbf{x}^{\mathrm{T}}(\mathbf{y} + \Delta\mathbf{y}) = (\mathbf{x} + \Delta\mathbf{x})^{\mathrm{T}}\mathbf{y}, \tag{12}$$

where

$$|\Delta\mathbf{y}| \le \gamma_{n+1}|\mathbf{y}|, \quad |\Delta\mathbf{x}| \le \gamma_{n+1}|\mathbf{x}|, \quad \gamma_{n+1} = \frac{(n+1)u}{1-(n+1)u}.$$

We assume that $nu \ll 1$. So we can think of γ_n as approximately nu.

4. LANCZOS TRIDIAGONALIZATION

Analogous to the analysis of the conventional Lanczos tridiagonalization,[5] we shall show that column vectors $\mathbf{q}_j$ can lose c-orthogonality. Let

$$\widehat{\alpha}_j = fl(\widehat{\mathbf{q}}_j^{\mathrm{T}}\widehat{\mathbf{v}}_j) = \widehat{\mathbf{q}}_j^{\mathrm{T}}(\widehat{\mathbf{v}}_j + \Delta\check{\mathbf{v}})$$

for some $\Delta\check{\mathbf{v}}$. It follows immediately from (12) that $\|\Delta\check{\mathbf{v}}\|_2 \le \gamma_{n+1}\|\widehat{\mathbf{v}}_j\|_2$. Thus,

$$\widehat{\alpha}_j = \widehat{\mathbf{q}}_j^{\mathrm{T}}\widehat{\mathbf{v}}_j + \Delta\alpha, \quad |\Delta\alpha| \le \gamma_{n+1}\|\widehat{\mathbf{q}}_j\|_2\,\|\widehat{\mathbf{v}}_j\|_2, \tag{13}$$

and

$$|\widehat{\alpha}_j| \le (1+\gamma_{n+1})\|\widehat{\mathbf{q}}_j\|_2\,\|\widehat{\mathbf{v}}_j\|_2. \tag{14}$$

If

$$\widehat{\mathbf{r}}_j = fl(\widehat{\mathbf{v}}_j - \widehat{\alpha}_j\widehat{\mathbf{q}}_j) \approx (\widehat{\mathbf{v}}_j + \Delta\mathbf{v}) - \widehat{\alpha}_j(\widehat{\mathbf{q}}_j + \Delta\mathbf{q}) \tag{15}$$

for some $\Delta\mathbf{v}$ and $\Delta\mathbf{q}$, then (9) and (10) show that

$$\|\Delta\mathbf{q}\|_2 \le \sqrt{2}\,\gamma_2\|\widehat{\mathbf{q}}_j\|_2, \qquad \|\Delta\mathbf{v}\|_2 \le u\|\widehat{\mathbf{v}}_j - \widehat{\alpha}_j\widehat{\mathbf{q}}_j\|_2.$$

Denoting $\Delta\mathbf{r} = \Delta\mathbf{v} - \widehat{\alpha}_j\Delta\mathbf{q}$ as the first order error in $\widehat{\mathbf{r}}_j$ and using (14), we get

$$\|\Delta\mathbf{r}\|_2 \le u\|\widehat{\mathbf{v}}_j - \widehat{\alpha}_j\widehat{\mathbf{q}}_j\|_2 + |\widehat{\alpha}_j|\sqrt{2}\,\gamma_2\|\widehat{\mathbf{q}}_j\|_2 \le (u + (\sqrt{2}\,\gamma_2 + u)\|\widehat{\mathbf{q}}_j\|_2^2)\|\widehat{\mathbf{v}}_j\|_2. \tag{16}$$

Now, from (11), we have

$$\widehat{\mathbf{q}}_{j+1} = fl(\widehat{\mathbf{r}}_j/\widehat{\beta}_j) = (\widehat{\mathbf{r}}_j + \Delta\check{\mathbf{r}})/\widehat{\beta}_j, \qquad \|\Delta\check{\mathbf{r}}\|_2 \le \sqrt{2}\,\gamma_4\|\widehat{\mathbf{r}}_j\|_2;$$

that is,

$$\widehat{\beta}_j\widehat{\mathbf{q}}_{j+1} = \widehat{\mathbf{r}}_j + \Delta\check{\mathbf{r}}, \qquad \|\Delta\check{\mathbf{r}}\|_2 \le \sqrt{2}\,\gamma_4\|\widehat{\mathbf{r}}_j\|_2. \tag{17}$$

To check c-orthogonality, we multiply $\widehat{\mathbf{q}}_j^{\mathrm{T}}$ on the both sides of the above equation:

$$\widehat{\beta}_j\widehat{\mathbf{q}}_j^{\mathrm{T}}\widehat{\mathbf{q}}_{j+1} = \widehat{\mathbf{q}}_j^{\mathrm{T}}(\widehat{\mathbf{r}}_j + \Delta\check{\mathbf{r}}) \approx \widehat{\mathbf{q}}_j^{\mathrm{T}}\widehat{\mathbf{v}}_j - \widehat{\alpha}_j\widehat{\mathbf{q}}_j^{\mathrm{T}}\widehat{\mathbf{q}}_j + \widehat{\mathbf{q}}_j^{\mathrm{T}}(\Delta\mathbf{r} + \Delta\check{\mathbf{r}}).$$

Since $\widehat{\mathbf{q}}_j$ is normalized and $\widehat{\mathbf{q}}_j^{\mathrm{T}}\widehat{\mathbf{v}}_j = \widehat{\alpha}_j - \Delta\alpha$ from (13), we get

$$\widehat{\beta}_j\widehat{\mathbf{q}}_j^{\mathrm{T}}\widehat{\mathbf{q}}_{j+1} \approx -\Delta\alpha + \widehat{\mathbf{q}}_j^{\mathrm{T}}(\Delta\mathbf{r} + \Delta\check{\mathbf{r}}). \tag{18}$$

Next we derive the bound for $\|\Delta\check{\mathbf{r}}\|_2$ by using (17), (15), and (14),

$$\|\Delta\check{\mathbf{r}}\|_2 \le \sqrt{2}\,\gamma_4\|\widehat{\mathbf{v}}_j - \widehat{\alpha}_j\widehat{\mathbf{q}}_j\|_2 \le \sqrt{2}\,\gamma_4(1 + \|\widehat{\mathbf{q}}_j\|_2^2)\|\widehat{\mathbf{v}}_j\|_2. \tag{19}$$

It follows from (16) and (19) that

$$\begin{aligned}
\|\Delta\mathbf{r} + \Delta\check{\mathbf{r}}\|_2 &\le \|\Delta\mathbf{r}\|_2 + \|\Delta\check{\mathbf{r}}\|_2 \\
&\le \left((\sqrt{2}\,\gamma_2 + u)\|\widehat{\mathbf{q}}_j\|_2^2 + u\right)\|\widehat{\mathbf{v}}_j\|_2 + \sqrt{2}\,\gamma_4(1 + \|\widehat{\mathbf{q}}_j\|_2^2)\|\widehat{\mathbf{v}}_j\|_2 \\
&= \left((\sqrt{2}\,\gamma_2 + \sqrt{2}\,\gamma_4 + u)\|\widehat{\mathbf{q}}_j\|_2^2 + (\sqrt{2}\,\gamma_4 + u)\right)\|\widehat{\mathbf{v}}_j\|_2.
\end{aligned}$$

Finally, (13) and (18) imply that

$$\begin{aligned}
|\widehat{\beta}_j\widehat{\mathbf{q}}_j^{\mathrm{T}}\widehat{\mathbf{q}}_{j+1}| &\approx |\Delta\alpha| + \|\Delta\mathbf{r} + \Delta\check{\mathbf{r}}\|_2\|\widehat{\mathbf{q}}_j\|_2 \\
&\le \left((\gamma_{n+1} + \sqrt{2}\,\gamma_4 + u) + (\sqrt{2}\,\gamma_2 + \sqrt{2}\,\gamma_4 + u)\|\widehat{\mathbf{q}}_j\|_2^2\right)\|\widehat{\mathbf{v}}_j\|_2\,\|\widehat{\mathbf{q}}_j\|_2.
\end{aligned}$$

Using the approximation $\gamma_n \approx nu$ when $nu \ll 1$ and ignoring the moderate constant coefficients, we obtain

$$|\widehat{\mathbf{q}}_j^{\mathrm{T}}\widehat{\mathbf{q}}_{j+1}| \le \left(\frac{(n + \|\widehat{\mathbf{q}}_j\|_2^2)\|\widehat{\mathbf{q}}_j\|_2\,\|\widehat{\mathbf{v}}_j\|_2}{|\widehat{\beta}_j|}\right)u. \tag{20}$$

The above inequality says that $\widehat{\mathbf{q}}_j$ can lose c-orthogonality when either $|\widehat{\beta}_j|$ is small or $\|\widehat{\mathbf{q}}_j\|_2$ is large (recall that $\widehat{\mathbf{q}}_j$ and $\widehat{\beta}_j$ denote computed results). Hence the loss is due to either the cancellation in computing β_j or the growth in $\|\widehat{\mathbf{q}}_j\|_2$, and not the accumulation of roundoff errors. In the real case, $\|\widehat{\mathbf{q}}_j\|_2 \approx 1$ and (20) is consistent with the results in Paige,[5] whereas in the complex case, the norm $\|\widehat{\mathbf{q}}_j\|_2$ can be arbitrarily large. This is an additional factor that causes the loss of c-orthogonality.

5. HANKEL MATRIX-VECTOR MULTIPLICATION

The only computation in the fast Hankel matrix-vector multiplication is

$$\mathbf{y} = \mathrm{ifft}(\mathrm{fft}(\widehat{\mathbf{c}}).*\mathrm{fft}(\widehat{\mathbf{w}})).$$

Denoting the computed $\mathbf{y}$ as

$$\widehat{\mathbf{y}} = fl(\mathrm{ifft}(\mathrm{fft}(\widehat{\mathbf{c}}).*\mathrm{fft}(\widehat{\mathbf{w}}))),$$

we consider the computation of $\mathrm{fft}(\widehat{\mathbf{w}})$. Assume that the weights ω_k^j in the FFT are precalculated and that the computed weights $\widehat{\omega}_k^j$ satisfy

$$\widehat{\omega}_k^j = \omega_k^j + \epsilon_{kj},$$

where

$$|\epsilon_{kj}| \le \mu$$

for all j and k. Depending on the method that computes the weights, we can pick

$$\mu = cu, \quad \mu = cu\log j, \quad \text{or} \quad \mu = cuj,$$

where c denotes a constant dependent on the method.[6] Following Higham[2] we let

$$\mathbf{x} = \mathrm{fft}(\widehat{\mathbf{w}}) \quad \text{and} \quad \widehat{\mathbf{x}} = fl(\mathrm{fft}(\widehat{\mathbf{w}})).$$

We have

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \le \sqrt{2n-1}\left(\frac{\eta\log_2(2n-1)}{1-\eta\log_2(2n-1)}\right)\|\mathbf{x}\|_2,$$

where

$$\eta = \mu + \gamma_4(1+\mu).$$

Approximately,

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \le \left(\frac{\sqrt{2n}\,\eta\log_2 n}{1-\eta\log_2 n}\right)\|\mathbf{x}\|_2. \tag{21}$$

Similarly, let

$$\mathbf{v} = \mathrm{fft}(\widehat{\mathbf{c}}) \quad \text{and} \quad \widehat{\mathbf{v}} = fl(\mathrm{fft}(\widehat{\mathbf{c}})).$$

Then

$$\|\mathbf{v} - \widehat{\mathbf{v}}\|_2 \le \left(\frac{\sqrt{2n}\,\eta\log_2 n}{1-\eta\log_2 n}\right)\|\mathbf{v}\|_2. \tag{22}$$

Next, we consider the componentwise multiplication $\mathbf{z} = \widehat{\mathbf{v}}.*\widehat{\mathbf{x}}$. Denoting the computed $\mathbf{z}$ as

$$\widehat{\mathbf{z}} = fl(\widehat{\mathbf{v}}.*\widehat{\mathbf{x}}),$$

we have approximately

$$\|\mathbf{z} - \widehat{\mathbf{z}}\|_2 \le u\,\|\mathbf{z}\|_2. \tag{23}$$

Finally, we consider

$$\mathbf{y} = \mathrm{ifft}(\widehat{\mathbf{z}}) \quad \text{and} \quad \widehat{\mathbf{y}} = fl(\mathrm{ifft}(\widehat{\mathbf{z}})).$$

Since F_n is the FFT matrix, we have

$$F_n^{-1} = n^{-1}F_n.$$

Thus, similar to (21) and (22), we get

$$\|\mathbf{y} - \widehat{\mathbf{y}}\|_2 \le \left(\frac{\sqrt{2n}\,\eta\log_2 n}{1-\eta\log_2 n}\right)\|\mathbf{y}\|_2. \tag{24}$$

As $n^{-1/2}F_n$ is unitary, it follows that

$$\|\mathbf{y}\|_2 = \|\text{ifft}(\widehat{\mathbf{z}})\|_2 = \|F_{2n-1}\widehat{\mathbf{z}}\|_2/(2n-1) = \|\widehat{\mathbf{z}}\|_2/\sqrt{2n-1} \approx \|\widehat{\mathbf{z}}\|_2/\sqrt{2n}.$$

Consequently, using (23) and (24), we get

$$\|\mathbf{y}-\widehat{\mathbf{y}}\|_2 \le \left(\frac{\eta\log_2 n}{1-\eta\log_2 n}\right)\|\widehat{\mathbf{z}}\|_2 \le \left(\frac{\eta\log_2 n}{1-\eta\log_2 n}\right)(1+u)\|\mathbf{z}\|_2.$$

Ignoring the second and higer order terms of u, we have

$$\|\mathbf{y}-\widehat{\mathbf{y}}\|_2 \le \left(\frac{\eta\log_2 n}{1-\eta\log_2 n}\right)\|\widehat{\mathbf{v}}.*\widehat{\mathbf{x}}\|_2. \tag{25}$$

Now, it remains to express $\|\widehat{\mathbf{v}}.*\widehat{\mathbf{x}}\|_2$ in terms of $\|\widehat{\mathbf{c}}\|_2$ and $\|\widehat{\mathbf{w}}\|_2$. Using this inequality:

$$\|\mathbf{a}.*\mathbf{b}\|_2 \le \max_i |a_i|\,\|\mathbf{b}\|_2 \le \|\mathbf{a}\|_2\|\mathbf{b}\|_2$$

and ignoring the second and higher order terms of u, we obtain

$$\begin{aligned}
\|\widehat{\mathbf{v}}.*\widehat{\mathbf{x}}\|_2 &\le \|\mathbf{v}.*\mathbf{x}\|_2 + \|(\mathbf{v}-\widehat{\mathbf{v}}).*\mathbf{x}\|_2 + \|\widehat{\mathbf{v}}.*(\mathbf{x}-\widehat{\mathbf{x}})\|_2 \\
&\le \|\mathbf{v}\|_2\,\|\mathbf{x}\|_2 + \|\mathbf{v}-\widehat{\mathbf{v}}\|_2\,\|\mathbf{x}\|_2 + \|\widehat{\mathbf{v}}\|_2\,\|\mathbf{x}-\widehat{\mathbf{x}}\|_2 \\
&\approx \|\mathbf{v}\|_2\,\|\mathbf{x}\|_2 + \|\mathbf{v}-\widehat{\mathbf{v}}\|_2\,\|\mathbf{x}\|_2 + \|\mathbf{v}\|_2\,\|\mathbf{x}-\widehat{\mathbf{x}}\|_2.
\end{aligned}$$

It follows from (21) and (22) that

$$\|\widehat{\mathbf{v}}.*\widehat{\mathbf{x}}\|_2 \le \left(1+2\sqrt{2n}\,\frac{\eta\log_2 n}{1-\eta\log_2 n}\right)\|\mathbf{v}\|_2\,\|\mathbf{x}\|_2. \tag{26}$$

Note that

$$\|\mathbf{v}\|_2 = \|\text{fft}(\widehat{\mathbf{c}})\|_2 \approx \sqrt{2n}\,\|\widehat{\mathbf{c}}\|_2$$

and

$$\|\mathbf{x}\|_2 \approx \sqrt{2n}\,\|\widehat{\mathbf{w}}\|_2.$$

Using (25) and (26) and ignoring the second and higher order terms, we obtain

$$\|\mathbf{y}-\widehat{\mathbf{y}}\|_2 \le \left(\frac{2n\eta\log_2 n}{1-\eta\log_2 n}\right)\|\widehat{\mathbf{c}}\|_2\,\|\widehat{\mathbf{w}}\|_2.$$

What does the above error bound tell us about the accuracy of the fast multiplication? For simplicity, we assume that the entries in the Hankel matrix H are about the same size. Recall that $\widehat{\mathbf{c}}$ consists of the first column and the last row of H. So

$$\|\widehat{\mathbf{c}}\|_2 \approx \sqrt{2/n}\,\|H\|_{\mathrm{F}},$$

and we conclude that

$$\|\mathbf{y}-\widehat{\mathbf{y}}\|_2 \le \left(\frac{2n\eta\sqrt{2}}{1-\eta\log_2 n}\right)\frac{\log_2 n}{\sqrt{n}}\,\|H\|_{\mathrm{F}}\,\|\widehat{\mathbf{w}}\|_2.$$

For conventional matrix-vector multiplication $H\mathbf{w}$, we have[2]

$$\|H\mathbf{w} - fl(H\mathbf{w})\|_2 \le \gamma_n\|H\|_2\,\|\mathbf{w}\|_2.$$

Since $\gamma_n \approx nu$ and η is of size u, our fast multiplication is more accurate than conventional multiplication by roughly a factor of $\log_2 n/\sqrt{n}$.

In summary, the fast Hankel matrix-vector multiplication scheme is stable. This is consistent with the error bound for the FFT, which says that the FFT has an error bound smaller than that for conventional multiplication by the same factor as the reduction in complexity of the method.[2]

6. CONDITION OF COMPLEX-ORTHOGONAL TRANSFORMATION

In this section, we derive the condition number of G of (8) and present a backward error analysis of an application of G. First, consider $\mathrm{cond}(G)$. We have

$$G^{\mathrm{H}}G = \begin{pmatrix} t & iv \\ -iv & t \end{pmatrix}, \tag{27}$$

where

$$t = |c|^2 + |s|^2 \quad \text{and} \quad v = 2\,\mathrm{Im}(\bar{c}s).$$

It can be verified that the two eigenvalues of $G^{\mathrm{H}}G$ are $t+v$ and $t-v$, and that

$$\mathrm{cond}(G) = \sqrt{\frac{t+|v|}{t-|v|}}.$$

Suppose that G is constructed to annihilate the second entry of $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$. Then in (8)

$$c = \xi\, x_1 \quad \text{and} \quad s = \xi\, x_2,$$

where

$$\xi = \frac{1}{|\mathbf{x}^{\mathrm{T}}\mathbf{x}|}.$$

Consequently, in (27)

$$t = \xi\, \mathbf{x}^{\mathrm{H}}\mathbf{x} \quad \text{and} \quad v = \xi\, \mathbf{x}^{\mathrm{H}} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \mathbf{x}.$$

The two eigenvalues $t \pm v$ are therefore

$$\xi\, \mathbf{x}^{\mathrm{H}} \begin{pmatrix} 1 & -i \\ i & 1 \end{pmatrix} \mathbf{x} \quad \text{and} \quad \xi\, \mathbf{x}^{\mathrm{H}} \begin{pmatrix} 1 & i \\ -i & 1 \end{pmatrix} \mathbf{x}.$$

The condition number of G can be arbitrarily large for some complex $\mathbf{x}$. For example, when $\mathbf{x} = (1+\delta, i)^{\mathrm{T}}$, the two eigenvalues are

$$\frac{|\delta|}{|2+\delta|} \quad \text{and} \quad \frac{|2+\delta|}{|\delta|}.$$

This also shows that $\|G\|_2$ can be arbitrarily large. For any real $\mathbf{x} \neq 0$, however, $t = 1$, $v = 0$, and $\mathrm{cond}(G) = \|G\|_2 = 1$.

Now we consider the application of G to $\mathbf{y} = (y_1, y_2)^{\mathrm{T}}$. Let

$$\widehat{\mathbf{z}} = fl(G\mathbf{y}) = fl \begin{pmatrix} cy_1 + sy_2 \\ -sy_1 + cy_2 \end{pmatrix}.$$

From (9) and (10), we get

$$\widehat{\mathbf{z}} = \begin{pmatrix} (cy_1(1+\delta_1) + sy_2(1+\delta_2))(1+\delta_3) \\ (-sy_1(1+\delta_4) + cy_2(1+\delta_5))(1+\delta_6) \end{pmatrix},$$

where $|\delta_i| \leq \sqrt{2}\,\gamma_2$ for $i = 1,2,4,5$ and $|\delta_i| \leq u$ for $i = 3,6$. Using first-order error analysis, we derive

$$\widehat{\mathbf{z}} = (G + \Delta G)\mathbf{y} \quad \text{where} \quad \Delta G = \begin{pmatrix} c(\delta_1+\delta_3) & s(\delta_2+\delta_3) \\ -s(\delta_4+\delta_6) & s(\delta_5+\delta_6) \end{pmatrix}$$

and

$$|\Delta G| \leq (\sqrt{2}\,\gamma_2 + u)\,|G|.$$

This shows that the application of G is perfectly stable in the real case but can be unstable in the complex case because $|G|$ can be large for some complex $\mathbf{x}$.

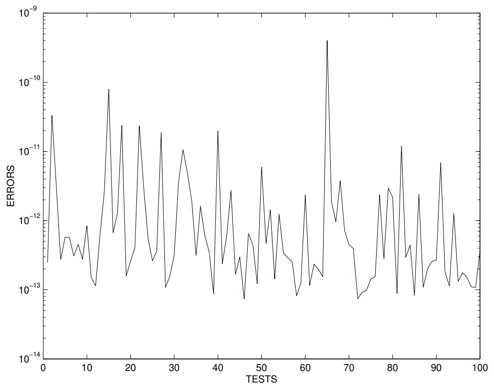

Figure 1. Errors in the eigenvalues of one hundred random Hankel matrices.

7. NUMERICAL EXPERIMENTS

We implemented the fast eigenvalue algorithm for Hankel matrices in MATLAB and tested the following examples on a SUN Ultra 1 station.

Example 1 We generated one hundred 20×20 random complex Hankel matrices. Each matrix was constructed by choosing two random complex vectors as its first column and as its last row. Both the real and imaginary parts of each component of the random vectors were uniformally distributed over $[-1, 1]$. For each matrix, we measured errors by assuming that the eigenvalues computed by the MATLAB function `eig` are exact. Let $\widehat{\lambda}$ and λ be the eigenvalues computed by our program and by MATLAB, respectively. Figure 1 plots the square roots of the sum of squares of the relative errors:

$$E_{\text{eig}} = \left(\sum_{i=1}^{20} \frac{|\widehat{\lambda}_i - \lambda_i|^2}{|\lambda_i|^2} \right)^{1/2}. \tag{28}$$

We observe that the relative accuracy is generally better than 10^{-12}.

Example 2 A rank deficient complex Hankel matrix H was generated using the Vandermonde decomposition:

$$H = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_k \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{n-1} & z_2^{n-1} & \cdots & z_k^{n-1} \end{pmatrix} \begin{pmatrix} a_1 & & & 0 \\ & a_2 & & \\ & & \ddots & \\ 0 & & & a_k \end{pmatrix} \begin{pmatrix} 1 & z_1 & \cdots & z_1^{n-1} \\ 1 & z_2 & \cdots & z_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & z_k & \cdots & z_k^{n-1} \end{pmatrix}. \tag{29}$$

The $n \times n$ matrix H has rank k. In this example, we chose $n = 10$, $k = 6$, z_i as random complex numbers with modulus one, and a_i as random real numbers uniformly distributed on $[-1, 1]$:

$$\mathbf{z} = \begin{pmatrix} 0.8585 - 0.5128i \\ 0.9915 - 0.1301i \\ 0.8308 + 0.5565i \\ -0.0900 - 0.9959i \\ 0.9855 - 0.1696i \\ 0.3677 + 0.9299i \end{pmatrix}, \qquad \mathbf{a} = \begin{pmatrix} 0.8436 \\ 0.4764 \\ -0.6475 \\ -0.1886 \\ 0.8709 \\ 0.8338 \end{pmatrix}.$$

Using (29), we generated the first column and the last row of a complex Hankel matrix. Then we slightly perturbed these two vectors by adding two small random noise vectors of size 10^{-6} with components normally distributed with zero mean. We obtained the following two vectors as the first column and the last row of our test Hankel matrix:

$$H_{1,:} = \begin{pmatrix} 2.1887 - 0.0000i \\ 1.8406 - 0.0394i \\ 1.0119 - 1.2191i \\ 0.4866 - 2.6229i \\ 0.9623 - 2.7117i \\ 1.7038 - 1.5199i \\ 1.2395 - 0.2055i \\ -0.2300 + 0.4271i \\ -0.8873 + 0.1759i \\ -0.1035 - 0.6176i \end{pmatrix}, \qquad H_{:,n} = \begin{pmatrix} -0.1035 - 0.6176i \\ 0.7279 - 1.0566i \\ 0.5042 - 0.7927i \\ -0.2653 - 0.8801i \\ -0.7093 - 2.1353i \\ -0.6196 - 3.3412i \\ -0.2362 - 2.8084i \\ -0.1845 - 0.9649i \\ -1.1269 + 0.3874i \\ -2.5246 + 0.6284i \end{pmatrix}^{\mathrm{T}}.$$

In the sixth and seventh Lanczos iterations, we got small subdiagonal elements:

$$\beta_6 = (3.4136 + 1.3181i) \times 10^{-4} \quad \text{and} \quad \beta_7 = (2.0738 + 7.3102i) \times 10^{-6}.$$

The complex-orthogonality of the matrix Q computed by the Lanczos method was lost:

$$\|Q^{\mathrm{T}}Q - I_{10}\|_{\mathrm{F}} = 2.1.$$

Our fast algorithm computed the eigenvalues

$$\begin{array}{r} -1.3168 - 9.1207i \\ 4.3390 - 7.2136i \\ -1.3928 + 6.1754i \\ -1.0186 + 0.9140i \\ 1.0442 - 0.3515i \\ -0.0061 + 0.0215i \\ 0.0000 - 0.0000i \\ 0.0000 + 0.0000i \\ 4.3588 - 7.1929i \\ -1.4280 - 9.2212i \end{array}$$

The last two eigenvalues are spurious. For comparison, MATLAB `eig` gave

$$\begin{array}{r} -1.3168 - 9.1207i \\ 4.3390 - 7.2136i \\ -1.3928 + 6.1754i \\ -1.0186 + 0.9140i \\ 1.0442 - 0.3515i \\ -0.0061 + 0.0215i \\ 0.0000 - 0.0000i \\ 0.0000 + 0.0000i \\ 0.0000 + 0.0000i \\ 0.0000 + 0.0000i \end{array}$$

The errors in the first six eigenvalues were in the magnitude of 10^{-13}. So, if we know the rank or an estimate k, we can stop the Lanczos process after k iterations. This leads to an algorithm for computing the k dominant eigenvalues of a complex Hankel matrix. In this example, we quit the Lanczos procedure after six iterations and computed the eigenvalues of the 6-by-6 triadiagonal matrix. All six computed eigenvalues were correct to at least four digits. The three largest eigenvalues were correct to at least nine digits.

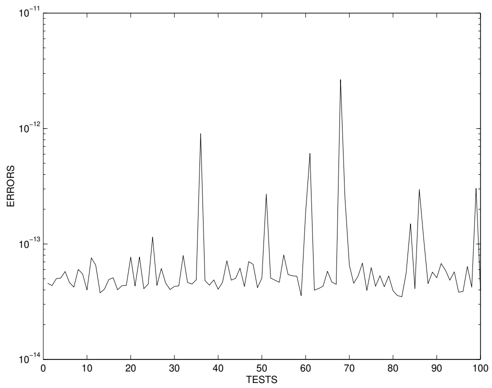

Figure 2. Errors in the eigenvalues of one hundred random Hermitian Toeplitz matrices.

Example 3 Our fast algorithm for complex Hankel matrices can be adapted for Hermitian Toeplitz matrices by incorporating the following simple changes:

- Replace the fast matrix-vector multiplication algorithm for Hankel matrices with one for Toeplitz matrices.
- Replace the modified Lanczos method with a conventional one for Hermitian matrices;
- Replace the modified QR method with a conventional one for Hermitian matrices.

We generated one hundred 20-element random complex vectors as the first columns of the Hermitian Toeplitz matrices. Figure 2 shows the error parameter E_{eig} as defined in (28). Not surprisingly, the errors here are smaller than those in Example 1; specifically, the accuracy is better than 10^{-13}.

REFERENCES

1. G. H. Golub and C. F. Van Loan. *Matrix Computations*. 3rd Ed., The Johns Hopkins University Press, Baltimore, MD, USA, 1996.
2. N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.
3. F. T. Luk and S. Qiao. Using Complex-Orthogonal Transformations to Diagonalize a Complex-Symmetric Matrix. In *Advanced Signal Processing Algorithms, Architectures, and Implementations VII*, Franklin T. Luk, Editor, Proceedings of SPIE Vol. 3162, pp. 418–425, 1997.
4. F. T. Luk and S. Qiao. A Fast Eigenvalue Algorithm for Hankel Matrices. In *Advanced Signal Processing Algorithms, Architectures, and Implementations VIII*, Franklin T. Luk, Editor, Proceedings of SPIE Vol. 3461, pp. 249–256, 1998.
5. C. C. Paige. Error Analysis of the Lanczos Algorithm for Tridiagonalizing a Symmetric Matrix. *J. Inst. Maths Applics*, Vol. 18, pp. 341–349, 1976.
6. C. F. Van Loan. *Computational Frameworks for the Fast Fourier Transform,* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.