MATHEMATICS OF COMPUTATION
Volume 77, Number 262, April 2008, Pages 1037–1056
S 0025-5718(07)02014-5
Article electronically published on October 1, 2007

THE GENERALIZED TRIANGULAR DECOMPOSITION

YI JIANG, WILLIAM W. HAGER, AND JIAN LI

Abstract. Given a complex matrix $\mathbf{H}$, we consider the decomposition $\mathbf{H} = \mathbf{QRP}^*$, where $\mathbf{R}$ is upper triangular and $\mathbf{Q}$ and $\mathbf{P}$ have orthonormal columns. Special instances of this decomposition include the singular value decomposition (SVD) and the Schur decomposition where $\mathbf{R}$ is an upper triangular matrix with the eigenvalues of $\mathbf{H}$ on the diagonal. We show that any diagonal for $\mathbf{R}$ can be achieved that satisfies Weyl's multiplicative majorization conditions:

$$\prod_{i=1}^{k} |r_i| \le \prod_{i=1}^{k} \sigma_i, \quad 1 \le k < K, \qquad \prod_{i=1}^{K} |r_i| = \prod_{i=1}^{K} \sigma_i,$$

where K is the rank of $\mathbf{H}$, σ_i is the i-th largest singular value of $\mathbf{H}$, and r_i is the i-th largest (in magnitude) diagonal element of $\mathbf{R}$. Given a vector $\mathbf{r}$ which satisfies Weyl's conditions, we call the decomposition $\mathbf{H} = \mathbf{QRP}^*$, where $\mathbf{R}$ is upper triangular with prescribed diagonal $\mathbf{r}$, the generalized triangular decomposition (GTD). A direct (nonrecursive) algorithm is developed for computing the GTD. This algorithm starts with the SVD and applies a series of permutations and Givens rotations to obtain the GTD. The numerical stability of the GTD update step is established. The GTD can be used to optimize the power utilization of a communication channel, while taking into account quality of service requirements for subchannels. Another application of the GTD is to inverse eigenvalue problems where the goal is to construct matrices with prescribed eigenvalues and singular values.

1. Introduction

Given a rank K matrix $\mathbf{H} \in \mathbb{C}^{m \times n}$, we consider the decomposition $\mathbf{H} = \mathbf{QRP}^*$ where $\mathbf{R}$ is a K by K upper triangular matrix, $\mathbf{Q}$ and $\mathbf{P}$ have orthonormal columns, and * denotes conjugate transpose. Special instances of this decomposition are, in chronological order:

(a) The *singular value decomposition* (SVD) [2, 19]

$$\mathbf{H} = \mathbf{V\Sigma W}^*,$$

where $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values on the diagonal.

(b) The *Schur decomposition* [22]

$$\mathbf{H} = \mathbf{QUQ}^*,$$

Received by the editor July 21, 2005 and, in revised form, June 15, 2006.
2000 *Mathematics Subject Classification.* Primary 15A23, 65F25, 94A11, 60G35.
Key words and phrases. Generalized triangular decomposition, geometric mean decomposition, matrix factorization, unitary factorization, singular value decomposition, Schur decomposition, MIMO systems, inverse eigenvalue problems.
This material is based on work supported by the National Science Foundation under Grants 0203270, 0619080, 0620286, and CCR-0097114.

where $\mathbf{U}$ is an upper triangular matrix with the eigenvalues of $\mathbf{H}$ on the diagonal.

(c) The QR *factorization* [5, 14]

$$\mathbf{H} = \mathbf{QR},$$

where $\mathbf{R}$ is upper triangular and $\mathbf{Q}$ is unitary (here $\mathbf{P} = \mathbf{I}$).

(d) The *complete orthogonal decomposition* [7, 10]

$$\mathbf{H} = \mathbf{Q}_2\mathbf{R}_2\mathbf{Q}_1^*,$$

where $\mathbf{H}^* = \mathbf{Q}_1\mathbf{R}_1$ is the QR factorization of $\mathbf{H}^*$ and $\mathbf{R}_1^* = \mathbf{Q}_2\mathbf{R}_2$ is the QR factorization of $\mathbf{R}_1^*$.

(e) The *geometric mean decomposition* (GMD) [15, 17, 20, 27]

$$\mathbf{H} = \mathbf{QRP}^*,$$

where $\mathbf{R}$ is upper triangular and the diagonal elements are the geometric mean of the positive singular values.

In this paper, we consider the general class of decompositions $\mathbf{H} = \mathbf{QRP}^*$, where the diagonal $\mathbf{r}$ of $\mathbf{R}$ is prescribed. We show that such a decomposition exists if $\mathbf{r}$ is "multiplicatively majorized" by the singular values of $\mathbf{H}$. More precisely, given two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, we write $\mathbf{a} \prec \mathbf{b}$ if

$$\prod_{i=1}^{k} |a_{[i]}| \le \prod_{i=1}^{k} |b_{[i]}| \quad \text{whenever} \quad 1 \le k \le n,$$

where "$[i]$" denotes the component of the vector with i-th largest magnitude. If $\mathbf{a} \prec \mathbf{b}$ and

$$\prod_{i=1}^{n} |a_i| = \prod_{i=1}^{n} |b_i|,$$

we write $\mathbf{a} \preceq \mathbf{b}$. We show that for any vector $\mathbf{r} \in \mathbb{C}^K$, the decomposition $\mathbf{H} = \mathbf{QRP}^*$ can be achieved if $\mathbf{r} \preceq \boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ is the vector consisting of the positive singular values of $\mathbf{H}$. We call this decomposition the generalized triangular decomposition (GTD) based on $\mathbf{r}$.

Since singular values are invariant under unitary transformations, it follows that $\mathbf{H}$ and $\mathbf{R}$ have the same singular values. Since $\mathbf{R}$ is upper triangular, its eigenvalues are the diagonal elements r_i, $1 \le i \le K$. By a theorem [24] of Weyl, $\mathbf{r} \preceq \boldsymbol{\sigma}$. An inverse result is given by Horn [12]: For any $\mathbf{r}$ for which $\mathbf{r} \preceq \boldsymbol{\sigma}$, there exists an upper triangular matrix $\mathbf{R}$ with diagonal elements r_i and singular values σ_i, $1 \le i \le K$. As a consequence of Horn's result, we show in Section 2 that for any $\mathbf{H} \in \mathbb{C}^{m\times n}$ of rank K and for any $\mathbf{r} \in \mathbb{C}^K$ with $\mathbf{r} \preceq \boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ is the vector of positive singular values for $\mathbf{H}$, there exist matrices $\mathbf{Q}$ and $\mathbf{P}$ with orthonormal columns such that $\mathbf{H} = \mathbf{QRP}^*$, where $\mathbf{R} \in \mathbb{C}^{K\times K}$ is upper triangular with diagonal equal to $\mathbf{r}$.

In Section 3 we give an algorithm for evaluating the GTD. Similar to our algorithm for the GMD, we start with the singular value decomposition, and apply a series of permutations and Givens rotations to obtain $\mathbf{H} = \mathbf{QRP}^*$. This is a direct method, in contrast to Chu's [4] recursive procedure for constructing matrices with prescribed eigenvalues and singular values based on Horn's divide and conquer proof of the sufficiency of Weyl's product inequalities. In Section 4, we give another view of the GTD update by expressing it in terms of unitary transformations applied to the original matrix as opposed to Givens rotations applied to the singular value

decomposition. Section 5 focuses on the numerical stability of the GTD update for inexact arithmetic. Since the rotations in the GTD update are expressed in terms of a ratio that reduces to zero over zero when two singular values coalesce, there is a potential for instability. We show that the GTD update is stable, even when singular values coalesce.

The GMD, where the diagonal of $\mathbf{R}$ is the geometric mean of the singular values of $\mathbf{H}$, is a solution of the following maximin problem, which arises when one tries to optimize the data throughput of a multiple-input multiple-output (MIMO) system [15, 17, 27]:

$$\begin{aligned} &\max_{\mathbf{Q},\mathbf{P}} \ \min \ \{r_{ii} : 1 \le i \le K\} \\ &\text{subject to} \quad \mathbf{QRP}^* = \mathbf{H},\ \mathbf{Q}^*\mathbf{Q} = \mathbf{I},\ \mathbf{P}^*\mathbf{P} = \mathbf{I}, \\ &\qquad\qquad r_{ij} = 0 \text{ for all } i > j,\ \mathbf{R} \in \mathbb{R}^{K\times K}. \end{aligned} \tag{1.1}$$

Here $\mathbf{H}$ is the "channel matrix," a matrix which describes the communication network. The matrices $\mathbf{P}$ and $\mathbf{Q}$ correspond to filters applied to the transmitted and received signals. The maximin problem (1.1) arises when we try to optimize the worst possible error rate. The maximum data throughput is achieved when the filters $\mathbf{P}$ and $\mathbf{Q}$ are chosen to make the smallest r_{ii} as large as possible. The GMD is a special case of the GTD since the vector $\mathbf{r}$ whose entries equal the geometric mean of the positive singular values of $\mathbf{H}$ is multiplicatively majorized by the singular values of $\mathbf{H}$.

In [17, 18, 26, 27], it is shown that the equal diagonal solution to (1.1) significantly improves the overall bit error rate performance while maximizing channel capacity and reducing the encoding/decoding complexity. But when different subchannels have different priorities and different quality of service (QoS) requirements, the objective function may be different from that in (1.1), and the optimal $\mathbf{R}$ may not have all diagonal elements equal. For example, when transmitting both audio and video data in a communication network, the video transmission may require greater accuracy than the audio transmission. In this case, smaller diagonal elements may be allowed for the audio (low accuracy) subchannels compared to the video (high accuracy) subchannels.

A specific application of the GTD to communication with QoS constraints is given in [16], where we study the optimization problem

$$\begin{aligned} &\min_{\mathbf{F}} \quad \text{tr } (\mathbf{FF}^*) \\ &\text{subject to} \quad \begin{pmatrix} \mathbf{HF} \\ \mathbf{I}_L \end{pmatrix} = \mathbf{QR} \\ &\qquad\qquad \text{diag}(\mathbf{R}) = \{\sqrt{1+\rho_i}\}_{i=1}^{L}. \end{aligned} \tag{1.2}$$

Here "tr" denotes the trace, $\mathbf{F} \in \mathbb{C}^{n\times L}$ is the precoder, $\mathbf{I}_L$ is the L by L identity matrix, the ρ_i, $1 \le i \le L$, are related to the specified subchannel capacities, and diag($\mathbf{R}$) denotes the vector formed by the diagonal of $\mathbf{R}$, the upper triangular factor in the QR decomposition of the "augmented matrix"

$$\mathbf{G}_a = \begin{pmatrix} \mathbf{HF} \\ \mathbf{I}_L \end{pmatrix}.$$

The cost function tr $(\mathbf{F}\mathbf{F}^*)$ corresponds to the power utilization of the precoder. The optimization problem amounts to finding the precoder which uses minimum power, while providing the specified subchannel capacities.

In [16] we obtain an explicit formula for the solution of (1.2) using the GTD. In related work [8], Guess considers the QoS problem for a code-division multiple-access (CDMA) system. His problem reduces to

$$\begin{aligned} \min_{\mathbf{F}} \quad & \text{tr } (\mathbf{F}\mathbf{F}^*) \\ \text{subject to} \quad & \mathbf{I} + \mathbf{F}^*\mathbf{F} = \mathbf{R}^*\mathbf{R} \\ & \text{diag}(\mathbf{R}) = \{\sqrt{1+\rho_i}\}_{i=1}^{L}, \end{aligned}$$

which is a special case of (1.2) corresponding to $\mathbf{H} = \mathbf{I}$. Guess gives an algorithm for solving this special case, as well as a recursive procedure for solving the more general problem (1.2). As explained in [16], there are several technical advantages to our GTD-based solution. One important advantage is that the GTD can be computed very efficiently by a direct algorithm (see the Matlab code posted on William Hager's web site). Another advantage is that our algorithm yields the matrix $\mathbf{Q}$, which is useful for communication applications. In contrast, Guess' algorithm does not construct $\mathbf{Q}$ explicitly.

Another application of the GTD is to the construction of matrices that possess a prescribed set of eigenvalues and singular values. As noted by Chu in [4], "Such a construction might be useful in designing matrices with desired spectral specifications. Many important properties, such as the conditioning of a matrix, are determined by eigenvalues or singular values." See [11, Chapter 28] for a "gallery of test matrices." In [4] Horn's proof of Weyl's product inequalities is developed into a recursive procedure SVD_EIG for generating a matrix with prescribed singular values and eigenvalues. In contrast, our algorithm for the GTD is a direct method based on a series of Givens rotations and permutations. Given the singular values $\boldsymbol{\sigma}$ and the eigenvalues $\boldsymbol{\lambda}$, with $\boldsymbol{\lambda} \preceq \boldsymbol{\sigma}$, the GTD generates $\mathbf{Q}\mathbf{R}\mathbf{P}^*$ where $\boldsymbol{\lambda}$ lies on the diagonal of $\mathbf{R}$ and the singular values of $\mathbf{R}$ are $\boldsymbol{\sigma}$. Comparisons with Chu's recursive algorithm are given in Section 6. Note that Chu's routine SVD_EIG does not generate an upper triangular matrix; hence, it could not be used to obtain the GTD.

2. Existence of GTD

The following result is due to Weyl [24] (also see [13, p. 171]):

Theorem 2.1. *If $\mathbf{A} \in \mathbb{C}^{n\times n}$ with eigenvalues $\boldsymbol{\lambda}$ and singular values $\boldsymbol{\sigma}$, then $\boldsymbol{\lambda} \preceq \boldsymbol{\sigma}$.*

The following result is due to Horn [12] (also see [13, p. 220]):

Theorem 2.2. *If $\mathbf{r} \in \mathbb{C}^n$ and $\boldsymbol{\sigma} \in \mathbb{R}^n$ with $\mathbf{r} \preceq \boldsymbol{\sigma}$, then there exists an upper triangular matrix $\mathbf{R} \in \mathbb{C}^{n\times n}$ with singular values σ_i, $1 \le i \le n$, and with $\mathbf{r}$ on the diagonal of $\mathbf{R}$.*

We now combine Theorems 2.1 and 2.2 to obtain:

Theorem 2.3. *Let $\mathbf{H} \in \mathbb{C}^{m\times n}$ have rank K with singular values $\sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_K > 0$. There exists an upper triangular matrix $\mathbf{R} \in \mathbb{C}^{K\times K}$ and matrices $\mathbf{Q}$ and $\mathbf{P}$ with orthonormal columns such that $\mathbf{H} = \mathbf{Q}\mathbf{R}\mathbf{P}^*$ if and only if $\mathbf{r} \preceq \boldsymbol{\sigma}$.*

Proof. If $\mathbf{H} = \mathbf{QRP}^*$, then the eigenvalues of $\mathbf{R}$ are its diagonal elements and the singular values of $\mathbf{R}$ coincide with those of $\mathbf{H}$. By Theorem 2.1, $\mathbf{r} \preceq \boldsymbol{\sigma}$. Conversely, suppose that $\mathbf{r} \preceq \boldsymbol{\sigma}$. Let $\mathbf{H} = \mathbf{V\Sigma W}^*$ be the singular value decomposition, where $\mathbf{\Sigma} \in \mathbb{R}^{K\times K}$. By Theorem 2.2, there exists an upper triangular matrix $\mathbf{R} \in \mathbb{C}^{K\times K}$ with the r_i on the diagonal and with singular values σ_i, $1 \le i \le K$. Let $\mathbf{R} = \mathbf{V}_0\mathbf{\Sigma W}_0^*$ be the singular value decomposition of $\mathbf{R}$. Substituting $\mathbf{\Sigma} = \mathbf{V}_0^*\mathbf{RW}_0$ in the singular value decomposition for $\mathbf{H}$, we have

$$\mathbf{H} = (\mathbf{VV}_0^*)\mathbf{R}(\mathbf{WW}_0^*)^*.$$

In other words, $\mathbf{H} = \mathbf{QRP}^*$ where $\mathbf{Q} = \mathbf{VV}_0^*$ and $\mathbf{P} = \mathbf{WW}_0^*$. □

3. The GTD algorithm

Given a matrix $\mathbf{H} \in \mathbb{C}^{m\times n}$ with rank K and with singular values $\sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_K > 0$, and given a vector $\mathbf{r} \in \mathbb{C}^K$ such that $\mathbf{r} \preceq \boldsymbol{\sigma}$, we now give an algorithm for computing the decomposition $\mathbf{H} = \mathbf{QRP}^*$, where $\mathbf{P}$ and $\mathbf{Q}$ have orthonormal columns and $\mathbf{R}$ is upper triangular with $\mathbf{r}$ on the diagonal. This algorithm for the GTD essentially yields a constructive proof of Theorem 2.2.

Let $\mathbf{V\Sigma W}^*$ be the singular value decomposition of $\mathbf{H}$, where $\mathbf{\Sigma}$ is a K by K diagonal matrix with the diagonal containing the positive singular values. We let $\mathbf{R}^{(L)} \in \mathbb{C}^{K\times K}$ denote an upper triangular matrix with the following properties:

(a) $r_{ij}^{(L)} = 0$ when $i > j$ or $j > i \ge L$. In other words, the trailing principal submatrix of $\mathbf{R}^{(L)}$, starting at row L and column L, is diagonal.
(b) If $\mathbf{r}^{(L)}$ denotes the diagonal of $\mathbf{R}^{(L)}$, then the first $L-1$ elements of $\mathbf{r}$ and $\mathbf{r}^{(L)}$ are equal. In other words, the leading diagonal elements of $\mathbf{R}^{(L)}$ match the prescribed leading elements of the vector $\mathbf{r}$.
(c) $\mathbf{r}_{L:K} \preceq \mathbf{r}_{L:K}^{(L)}$, where $\mathbf{r}_{L:K}$ denotes the subvector of $\mathbf{r}$ consisting of components L through K. In other words, the trailing diagonal elements of $\mathbf{R}^{(L)}$ multiplicatively majorize the trailing elements of the prescribed vector $\mathbf{r}$.

Initially, we set $\mathbf{R}^{(1)} = \mathbf{\Sigma}$. Clearly, (a)–(c) hold for $L = 1$. Proceeding by induction, suppose we have generated upper triangular matrices $\mathbf{R}^{(L)}$, $L = 1, 2, \ldots, k$, satisfying (a)–(c), and unitary matrices $\mathbf{Q}_L$ and $\mathbf{P}_L$, such that $\mathbf{R}^{(L+1)} = \mathbf{Q}_L^*\mathbf{R}^{(L)}\mathbf{P}_L$ for $1 \le L < k$. We now show how to construct unitary matrices $\mathbf{Q}_k$ and $\mathbf{P}_k$ such that $\mathbf{R}^{(k+1)} = \mathbf{Q}_k^*\mathbf{R}^{(k)}\mathbf{P}_k$, where $\mathbf{R}^{(k+1)}$ satisfies (a)–(c) for $L = k+1$.

Let p and q be defined as follows:

$$p = \arg \min_i \{|r_i^{(k)}| : k \le i \le K,\ |r_i^{(k)}| \ge |r_k|\}, \tag{3.1}$$

$$q = \arg \max_i \{|r_i^{(k)}| : k \le i \le K,\ |r_i^{(k)}| \le |r_k|,\ i \ne p\}, \tag{3.2}$$

where $r_i^{(k)}$ is the i-th element of $\mathbf{r}^{(k)}$. Since $\mathbf{r}_{k:K} \preceq \mathbf{r}_{k:K}^{(k)}$, there exists p and q satisfying (3.1) and (3.2). Let $\mathbf{\Pi}$ be the matrix corresponding to the symmetric permutation $\mathbf{\Pi}^*\mathbf{R}^{(k)}\mathbf{\Pi}$ which moves the diagonal elements $r_{pp}^{(k)}$ and $r_{qq}^{(k)}$ to the k-th and $(k+1)$-st diagonal positions respectively. Let $\delta_1 = r_{pp}^{(k)}$ and $\delta_2 = r_{qq}^{(k)}$ denote the new diagonal elements at locations k and $k+1$ associated with the permuted matrix $\mathbf{\Pi}^*\mathbf{R}^{(k)}\mathbf{\Pi}$.

Next, we construct unitary matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ by modifying the elements in the identity matrix that lie at the intersection of rows k and $k+1$ and columns k and $k+1$. We multiply the permuted matrix $\mathbf{\Pi}^*\mathbf{R}^{(k)}\mathbf{\Pi}$ on the left by $\mathbf{G}_2^*$ and

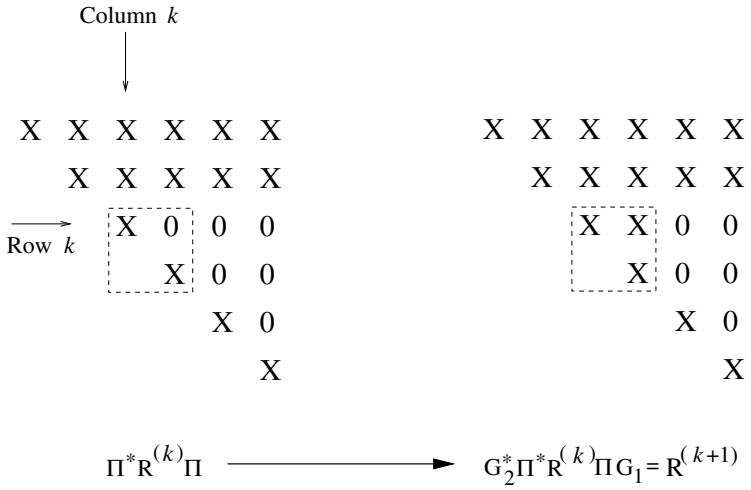

FIGURE 3.1. The operation displayed in (3.3)

on the right by $\mathbf{G}_1$. These multiplications will change the elements in the 2 by 2 submatrix at the intersection of rows k and $k+1$ with columns k and $k+1$. Our choice for the elements of $\mathbf{G}_1$ and $\mathbf{G}_2$ is shown below, where we focus on the relevant 2 by 2 submatrices of $\mathbf{G}_2^*$, $\mathbf{\Pi}^*\mathbf{R}^{(k)}\mathbf{\Pi}$, and $\mathbf{G}_1$:

$$\underset{(\mathbf{G}_2^*)}{\frac{r_k}{|r_k|^2}\begin{bmatrix} c\delta_1^* & s\delta_2^* \\ -s\delta_2 & c\delta_1 \end{bmatrix}} \quad \underset{(\mathbf{\Pi}^*\mathbf{R}^{(k)}\mathbf{\Pi})}{\begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix}} \quad \underset{(\mathbf{G}_1)}{\begin{bmatrix} c & -s \\ s & c \end{bmatrix}} = \underset{(\mathbf{R}^{(k+1)})}{\begin{bmatrix} r_k & x \\ 0 & y \end{bmatrix}}. \tag{3.3}$$

If $|\delta_1| = |\delta_2| = |r_k|$, we take $c = 1$ and $s = 0$; if $|\delta_1| \neq |\delta_2|$, we take

$$c = \sqrt{\frac{|r_k|^2 - |\delta_2|^2}{|\delta_1|^2 - |\delta_2|^2}} \quad \text{and} \quad s = \sqrt{1 - c^2}. \tag{3.4}$$

In either case,

$$x = \frac{sc(|\delta_2|^2 - |\delta_1|^2)r_k}{|r_k|^2} \quad \text{and} \quad y = \frac{\delta_1\delta_2 r_k}{|r_k|^2}. \tag{3.5}$$

Figure 3.1 depicts the transformation from $\mathbf{\Pi}^*\mathbf{R}^{(k)}\mathbf{\Pi}$ to $\mathbf{G}_2^*\mathbf{\Pi}^*\mathbf{R}^{(k)}\mathbf{\Pi}\mathbf{G}_1$. The dashed box is the 2 by 2 submatrix displayed in (3.3). Notice that c and s, defined in (3.4), are real scalars chosen so that

$$c^2 + s^2 = 1 \quad \text{and} \quad c^2|\delta_1|^2 + s^2|\delta_2|^2 = |r_k|^2. \tag{3.6}$$

With these identities, the validity of (3.3) follows by direct computation. By the choice of p and q, we have

$$|\delta_2| \leq |r_k| \leq |\delta_1|. \tag{3.7}$$

If $|\delta_1| \neq |\delta_2|$, it follows from (3.7) that c and s are real nonnegative scalars. It can be checked that the 2 by 2 matrices in (3.3) associated with $\mathbf{G}_1$ and $\mathbf{G}_2^*$ are both unitary. Consequently, both $\mathbf{G}_1$ and $\mathbf{G}_2$ are unitary. We define

$$\mathbf{R}^{(k+1)} = (\mathbf{\Pi}\mathbf{G}_2)^*\mathbf{R}^{(k)}(\mathbf{\Pi}\mathbf{G}_1) = \mathbf{Q}_k^*\mathbf{R}^{(k)}\mathbf{P}_k,$$

where $\mathbf{Q}_k = \mathbf{\Pi G}_2$ and $\mathbf{P}_k = \mathbf{\Pi G}_1$. By (3.3) and Figure 3.1, $\mathbf{R}^{(k+1)}$ has properties (a) and (b) for $L = k+1$. Now consider property (c).

We write $\mathbf{a} \sim \mathbf{b}$ if $\mathbf{a}$ and $\mathbf{b}$ are equal after a suitable reordering of the components. Let $\mathbf{a}$, $\mathbf{b}$, $\mathbf{a}^+$, and $\mathbf{b}^+$ be vectors whose components are ordered in decreasing magnitude, and which satisfy

$$\mathbf{a} \sim \mathbf{r}_{k:K}, \quad \mathbf{b} \sim \mathbf{r}^{(k)}_{k:K}, \quad \mathbf{a}^+ \sim \mathbf{r}_{k+1:K}, \quad \text{and} \quad \mathbf{b}^+ \sim \mathbf{r}^{(k+1)}_{k+1:K}. \tag{3.8}$$

Thus a_i is the i-th largest (in magnitude) component of $\mathbf{r}_{k:K}$. By the induction hypothesis, we have $\mathbf{a} \preceq \mathbf{b}$. To establish (c), we need to show that $\mathbf{a}^+ \preceq \mathbf{b}^+$. Let the index s be chosen so that $a_s = r_k$, and let the index t be chosen so that

$$|b_t| \geq |r_k| \geq |b_{t+1}|. \tag{3.9}$$

By the definition of p and q, $r^{(k)}_{pp} = b_t$ and $r^{(k)}_{qq} = b_{t+1}$. As seen in (3.8), $\mathbf{a}^+$ is obtained from $\mathbf{a}$ by deleting $a_s = r_k$. The vector $\mathbf{r}^{(k+1)}$ is obtained from $\mathbf{r}^{(k)}$ by a unitary transformation that changes the value of two elements. In particular, $\mathbf{b}^+$ is obtained from $\mathbf{b}$ by replacing the adjacent pair b_t and b_{t+1} by

$$y = \frac{b_t b_{t+1} r_k}{|r_k|^2}.$$

By (3.9) $|b_t| \geq |y| \geq |b_{t+1}|$. Consequently,

$$b_t^+ = y. \tag{3.10}$$

We partition the proof of (c) into 2 cases.

Case 1: $s \leq t$. Since $a_i^+ \leq a_i$ for all i, $\mathbf{a} \preceq \mathbf{b}$, and $b_i = b_i^+$ for $1 \leq i < t$, we have

$$\mathbf{a}^+_{1:t-1} \prec \mathbf{a}_{1:t-1} \prec \mathbf{b}_{1:t-1} = \mathbf{b}^+_{1:t-1}. \tag{3.11}$$

For $j > t \geq s$, it follows from the induction hypothesis and the connection between $\mathbf{a}$ and $\mathbf{a}^+$ that

$$|r_k| \prod_{i=1}^{j-1} |a_i^+| = |a_s| \prod_{i=1}^{j-1} |a_i^+| = \prod_{i=1}^{j} |a_i| \leq \prod_{i=1}^{j} |b_i|. \tag{3.12}$$

Since $\mathbf{G}_1$ and $\mathbf{G}_2$ are unitary, the determinant of (3.3) gives

$$|\delta_1 \delta_2| = |r^{(k)}_p r^{(k)}_q| = |b_t b_{t+1}| = |r_k y| = |r_k b_t^+|, \tag{3.13}$$

where the last equality in (3.13) comes from (3.10). Hence, for $j > t$, it follows that

$$\begin{aligned} \prod_{i=1}^{j} |b_i| &= |b_t||b_{t+1}| \left(\prod_{i=1}^{t-1} |b_i| \right) \left(\prod_{i=t+2}^{j} |b_i| \right) \\ &= |r_k||b_t^+| \left(\prod_{i=1}^{t-1} |b_i^+| \right) \left(\prod_{i=t+1}^{j-1} |b_i^+| \right) = |r_k| \prod_{i=1}^{j-1} |b_i^+|. \end{aligned} \tag{3.14}$$

Combining (3.11), (3.12), and (3.14), we have $\mathbf{a}^+ \preceq \mathbf{b}^+$.

Case 2: $s > t$. As before, (3.11) holds. For $t < j < s$, we have

$$\prod_{i=1}^{j} |a_i^+| = \prod_{i=1}^{j} |a_i| \leq \prod_{i=1}^{j} |b_i| = |r_k| \prod_{i=1}^{j-1} |b_i^+|,$$

where the first equality comes from the relation $j < s$, the middle inequality is the induction hypothesis, and the last equality is (3.14). Rearranging this gives

$$\left(\frac{|a_j|}{|r_k|}\right)\prod_{i=1}^{j-1}|a_i^+| \le \prod_{i=1}^{j-1}|b_i^+|. \tag{3.15}$$

Since $|a_j|/|r_k| = |a_j|/|a_s| \ge 1$ when $j < s$, we deduce from (3.15) that

$$\mathbf{a}^+_{1:j-1} \prec \mathbf{b}^+_{1:j-1}$$

when $j < s$. This also holds for $j \ge s$ due to (3.12) and (3.14). This completes the proof of (c).

Hence, there exists an upper triangular matrix $\mathbf{R}^{(K)}$, with $\mathbf{r}_{1:K-1}$ occupying the first $K-1$ diagonal elements, and unitary matrices $\mathbf{Q}_i$ and $\mathbf{P}_i$, $i = 1, 2, \ldots, K-1$, such that

$$\mathbf{R}^{(K)} = (\mathbf{Q}^*_{k-1}\ldots\mathbf{Q}^*_2\mathbf{Q}^*_1)\mathbf{\Sigma}(\mathbf{P}_1\mathbf{P}_2\ldots\mathbf{P}_{k-1}). \tag{3.16}$$

Equating determinants in (3.16) and utilizing the identity $r_i^{(k)} = r_i$ for $1 \le i \le K-1$, we have

$$\prod_{i=1}^{K}|r_i^{(K)}| = \frac{|r_K^{(K)}|}{|r_K|}\left(\prod_{i=1}^{K}|r_i|\right) = \prod_{i=1}^{K}\sigma_i = \prod_{i=1}^{K}|r_i|,$$

where the last equality is due to the assumption $\mathbf{r} \preceq \boldsymbol{\sigma}$. It follows that $|r_K^{(K)}| = |r_K|$. Let $\mathbf{C}$ be the diagonal matrix obtained by replacing the (K,K) element of the identity matrix by $r_K^{(K)}/r_K$. The matrix $\mathbf{C}$ is unitary since $|r_k|/|r_K^{(K)}| = 1$. The matrix

$$\mathbf{R} = \mathbf{C}^*\mathbf{R}^{(K)} \tag{3.17}$$

has diagonal equal to $\mathbf{r}$ due to the choice of $\mathbf{C}$.

Combining (3.16) and (3.17) with the singular value decomposition $\mathbf{H} = \mathbf{V\Sigma W}^*$ gives

$$\mathbf{H} = \mathbf{V}\mathbf{Q}_1\mathbf{Q}_2\ldots\mathbf{Q}_{k-1}\mathbf{C}\mathbf{R}\mathbf{P}^*_{k-1}\ldots\mathbf{P}^*_2\mathbf{P}^*_1\mathbf{W}^*.$$

Hence, we have obtained the GTD with

$$\mathbf{Q} = \mathbf{V}\left(\prod_{i=1}^{K-1}\mathbf{Q}_i\right)\mathbf{C} \quad \text{and} \quad \mathbf{P} = \mathbf{W}\left(\prod_{i=1}^{K-1}\mathbf{P}_i\right).$$

Finally, note that if $\mathbf{r}$ is real, then $\mathbf{G}_1$ and $\mathbf{G}_2$ are real, which implies $\mathbf{R}$ is real.

We summarize the steps of the GTD algorithm as follows. To make it easier to distinguish between the elements of the matrix $\mathbf{R}$ and the elements of the given diagonal vector $\mathbf{r}$, we use R_{ij} to denote the (i,j) element of $\mathbf{R}$ and r_i to denote the i-th element of $\mathbf{r}$.

1. Let $\mathbf{H} = \mathbf{V\Sigma W}^*$ be the singular value decomposition of $\mathbf{H}$, and suppose we are given $\mathbf{r} \in \mathbb{C}^K$ with $\mathbf{r} \preceq \boldsymbol{\sigma}$. Initialize $\mathbf{Q} = \mathbf{V}$, $\mathbf{P} = \mathbf{W}$, $\mathbf{R} = \mathbf{\Sigma}$, and $k = 1$.
2. Let p and q be defined as follows:

$$\begin{aligned} p &= \arg\ \min_i\{|R_{ii}| : k \le i \le K,\ |R_{ii}| \ge |r_k|\}, \\ q &= \arg\ \max_i\{|R_{ii}| : k \le i \le K,\ |R_{ii}| \le |r_k|,\ i \ne p\}. \end{aligned}$$

In $\mathbf{R}$, $\mathbf{P}$, and $\mathbf{Q}$, perform the following exchanges:

$$\begin{aligned}(R_{kk}, R_{k+1,k+1}) &\leftrightarrow (R_{pp}, R_{qq}),\\ (\mathbf{R}_{1:k-1,k}, \mathbf{R}_{1:k-1,k+1}) &\leftrightarrow (\mathbf{R}_{1:k-1,p}, \mathbf{R}_{1:k-1,q}),\\ (\mathbf{P}_{:,k}, \mathbf{P}_{:,k+1}) &\leftrightarrow (\mathbf{P}_{:,p}, \mathbf{P}_{:,q}),\\ (\mathbf{Q}_{:,k}, \mathbf{Q}_{:,k+1}) &\leftrightarrow (\mathbf{Q}_{:,p}, \mathbf{Q}_{:,q}).\end{aligned}$$

3. Construct the matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ shown in (3.3). Replace $\mathbf{R}$ by $\mathbf{G}_2^*\mathbf{R}\mathbf{G}_1$, replace $\mathbf{Q}$ by $\mathbf{Q}\mathbf{G}_2$, and replace $\mathbf{P}$ by $\mathbf{P}\mathbf{G}_1$.
4. If $k = K - 1$, then go to step 5. Otherwise, replace k by $k + 1$ and go to step 2.
5. Multiply column K of $\mathbf{Q}$ by R_{KK}/r_K; replace R_{KK} by r_K. The product $\mathbf{Q}\mathbf{R}\mathbf{P}^*$ is the GTD of $\mathbf{H}$ based on $\mathbf{r}$.

The numerical stability of this algorithm is analyzed in Section 5. In particular, the division by the possibly small denominator in (3.4) is safe, and the algorithm is stable. A MATLAB implementation of our GTD algorithm is posted on the web site of William Hager. Given the SVD, this algorithm for the GTD requires $O((m + n)K)$ flops. For comparison, reduction of $\mathbf{H}$ to bidiagonal form by the Golub-Kahan bidiagonalization scheme [6] (also see [7, 9, 23, 25]), often the first step in the computation of the SVD, requires $O(mnK)$ flops.

4. The GTD update

In this section, we give the rationale behind the GTD update (3.3). The prescribed diagonal element r_k satisfies the relation $|\delta_1| \geq |r_k| \geq |\delta_2|$. The first column of $\mathbf{G}_1$ is chosen so that the vector

$$\mathbf{p} = \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} \begin{bmatrix} c \\ s \end{bmatrix}$$

has length equal to $|r_k|$. When $c = 1$, $\mathbf{p}$ has length $|\delta_1|$, and when $s = 1$, $\mathbf{p}$ has length $|\delta_2|$. Hence, as (c, s) travels along the unit circle from $(1, 0)$ to $(0, 1)$, there exists a point where the length of $\mathbf{p}$ is $|r_k|$. The second column of $\mathbf{G}_1$ is chosen to be orthogonal to the first column of $\mathbf{G}_1$. The second column of $\mathbf{G}_2$ is also chosen to be orthogonal to $\mathbf{p}$, while the first column of $\mathbf{G}_2$ is orthogonal to the second column of $\mathbf{G}_2$. Since the second column of $\mathbf{G}_2$ is perpendicular to $\mathbf{p}$, the $(k+1, k)$ element of $\mathbf{R}^{(k+1)}$ is 0. Since multiplication by $\mathbf{G}_2$ preserves length, the (k, k) element of $\mathbf{R}^{(k+1)}$ has length $|r_k|$. Finally, we multiply $\mathbf{G}_2$ by a complex scalar of magnitude 1 in order to make the (k, k) element of $\mathbf{R}^{(k+1)}$ equal to r_k.

In principle, the procedure outlined above could be applied to the entire matrix, rather than to the diagonal matrix in the SVD. That is, we first construct a unit vector $\mathbf{p}_1 \in \mathbb{C}^n$ such that $\|\mathbf{H}\mathbf{p}_1\| = |r_1|$. Let $\mathbf{P}_1$ be a unitary matrix with first column $\mathbf{p}_1$. The matrix $\mathbf{P}_1$ can be expressed in terms of a Householder reflection [9, p. 210]. Let $\mathbf{Q}_1$ be a unitary matrix with first column $(r_1/|r_1|^2)\mathbf{H}\mathbf{p}_1$. For these matrices, we have

$$\mathbf{Q}_1^*\mathbf{H}_1\mathbf{P}_1 = \begin{bmatrix} r_1 & \mathbf{z}_2 \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix},$$

where $\mathbf{H}_1 = \mathbf{H}$, $\mathbf{z}_2 \in \mathbb{C}^{n-1}$, and $\mathbf{H}_2 \in \mathbb{C}^{(m-1)\times(n-1)}$.

The reduction to triangular form would continue in this same way; after $k-1$ steps, we have

$$(\prod_{j=1}^{k-1} \mathbf{Q}_j)^* \mathbf{\Sigma} (\prod_{j=1}^{k-1} \mathbf{P}_j) = \begin{bmatrix} \mathbf{R}_k & \mathbf{Z}_k \\ \mathbf{0} & \mathbf{H}_k \end{bmatrix}, \tag{4.1}$$

where $\mathbf{R}_k$ is a k by k upper triangular matrix with r_1, r_2, ..., r_k on the diagonal, $\mathbf{Q}_j$ and $\mathbf{P}_j$ are unitary, and $\mathbf{0}$ denotes a matrix whose entries are all 0. In the next step, we take

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{P}} \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_k = \begin{bmatrix} \mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{Q}} \end{bmatrix}, \tag{4.2}$$

where $\mathbf{I}_k$ is a k by k identity matrix. The first column $\bar{\mathbf{p}}$ of $\bar{\mathbf{P}}$ is chosen so that $\|\mathbf{H}_k \bar{\mathbf{p}}\| = |r_{k+1}|$, while the first column of $\bar{\mathbf{Q}}$ is $(r_{k+1}/|r_{k+1}|^2)\mathbf{H}_k \bar{\mathbf{p}}$.

The vector $\bar{\mathbf{p}}$ may be generated by a Lanczos process (see [6] or [21, Chap. 13]). That is, we first compute unit vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ such that

$$\|\mathbf{H}_k \mathbf{v}_1\| \geq |r_{k+1}| \geq \|\mathbf{H}_k \mathbf{v}_2\|. \tag{4.3}$$

Let $\mathbf{v}(\theta)$ be the vector obtained by rotating $\mathbf{v}_1$ through an angle θ towards $\mathbf{v}_2$. By continuity of the norm, there exists a value of θ such that $\|\mathbf{H}_k \mathbf{v}(\theta)\| = |r_{k+1}|$. For the GMD, where all the elements of $\mathbf{r}$ equal the geometric mean of the positive singular values of $\mathbf{H}$, we establish in [15] the existence of vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ satisfying (4.3). For a general $\mathbf{r}$ satisfying $\mathbf{r} \preceq \boldsymbol{\sigma}$, the existence (or nonexistence) of $\mathbf{v}_1$ and $\mathbf{v}_2$ satisfying (4.3) is an open problem. Hence, the algorithm in (4.1)–(4.3) is conceptual in the sense that the existence of $\mathbf{v}_1$ and $\mathbf{v}_2$ satisfying (4.3) has only been established for the GMD.

5. The GTD algorithm with inexact arithmetic

The numerical stability of the GTD algorithm (the 5 steps summarized at the end of Section 3) hinges on the computation of the product (3.3), where c and s are given in (3.4). When δ_1 and δ_2 are close together, there is a large relative error in the evaluation of c in finite precision floating point arithmetic (see [9, Sect. 1-4]). In this section, we show that these large errors in the evaluation of c and s are harmless.

Following the notation in [11], we put a hat over a quantity to denote its computed, numerical value (a floating point number). We also let $\mathrm{fl}(\cdot)$ stand for the floating representation of an expression which is evaluated using floating point arithmetic. If an expression is not surrounded by $\mathrm{fl}(\cdot)$, then all the operations are done using exact arithmetic. The "unit roundoff" (or machine epsilon) is denoted u. Typically, u is on the order of 10^{-8} or 10^{-16} in single or double precision respectively. We assume that floating point arithmetic is performed in accordance with IEEE standard 754 [1]. If x, y, and z denote three floating point numbers, then some implications of the IEEE standard, which are used in our analysis, are the following:

- F1. If "op" denotes either $+, -, \times$, or $\div$, then $\mathrm{fl}(x \text{ op } y) = (x \text{ op } y)(1+\epsilon)$ where $|\epsilon| \leq u$.
- F2. $\mathrm{fl}(\sqrt{x}) = \sqrt{x}(1+\epsilon)$ where $|\epsilon| \leq u$.
- F3. If $x \geq y \geq 0$ and $x > 0$, then $0 \leq \mathrm{fl}(y/x) \leq 1$.
- F4. If $x \geq y \geq z$, then $\mathrm{fl}(x-z) \geq \mathrm{fl}(x-y) \geq 0$ and $\mathrm{fl}(x-z) \geq \mathrm{fl}(y-z) \geq 0$.
- F5. If $0 \leq x \leq 1$, then $0 \leq \mathrm{fl}(\sqrt{x}) \leq 1$.

In this section, let $\mathbf{G}_1$ and $\mathbf{G}_2$ denote the 2 by 2 matrices depicted in (3.3). The floating point versions $\widehat{\mathbf{G}}_1$ and $\widehat{\mathbf{G}}_2$ of these matrices are obtained as follows: First, the floating point representation $\hat{c}$ of c is formed by substituting floating point numbers δ_1, δ_2, and r_k in (3.4) and performing floating point arithmetic. Then $\hat{c}$ is inserted in the equation for s in (3.4) to obtain the floating point value $\hat{s}$. Finally, the floating point numbers $\hat{c}$ and $\hat{s}$, along with floating point arithmetic, are used to construct the matrices $\widehat{\mathbf{G}}_1$ and $\widehat{\mathbf{G}}_2$ in (3.3).

Our main result in this section concerns how close the matrices $\widehat{\mathbf{G}}_1$ and $\widehat{\mathbf{G}}_2$ are to unitary matrices, and how close the numerical version of the identity (3.3) agrees with the exact version. Our analysis uses the following notation: If $g(u)$ is a scalar-valued function of the unit roundoff and $M > 0$ is a scalar, then we write $g(u) = O(Mu)$ if

$$\limsup_{u \to 0} \quad \frac{|g(u)|}{Mu} \leq 1.$$

If $z = x + y\mathbf{i}$ is a complex, floating point number, then by (F1), we have

$$\begin{aligned} \mathrm{fl}(|z|^2) = \mathrm{fl}(x^2 + y^2) &= (x^2(1+\epsilon_1) + y^2(1+\epsilon_2))(1+\epsilon_3) \\ &= x^2 + y^2 + (\epsilon_1 + \epsilon_3)x^2 + (\epsilon_2 + \epsilon_3)y^2 + \epsilon_1\epsilon_3 x^2 + \epsilon_2\epsilon_3 y^2, \end{aligned}$$

where $|\epsilon_i| \leq u$ for $i = 1, 2, 3$. The $(1 + \epsilon_1)$ and $(1 + \epsilon_2)$ factors are due to the error in the floating point squaring of x and y. The $(1 + \epsilon_3)$ factor is due to the error in the floating point addition operation. It follows that

$$|\mathrm{fl}(x^2 + y^2) - (x^2 + y^2)| \leq (2u + u^2)(x^2 + y^2) = O(2u)(x^2 + y^2).$$

Hence, we have

$$\mathrm{fl}(|z|^2) = |z|^2(1 + O(2u)). \tag{5.1}$$

Let f_r^2, f_1^2, and f_2^2 denote the floating point representations of $|r_k|^2$, $|\delta_1|^2$, and $|\delta_2|^2$ respectively. Here the superscript 2 in f_r^2 does not mean that f_r is squared; rather, f_r^2 is notation for the floating point representation of $|r_k|^2$.

Lemma 5.1. *If $f_1^2 \geq f_r^2 \geq f_2^2$, $f_1^2 > f_2^2$, and the floating point arithmetic satisfies IEEE standard 754, then we have*

$$\hat{c}^2 + \hat{s}^2 = 1 + O(4u), \tag{5.2}$$

$$\hat{c}^2 = \left(\frac{f_r^2 - f_2^2}{f_1^2 - f_2^2}\right)(1 + O(5u)), \quad \text{and } 0 \leq \hat{s}, \hat{c} \leq 1. \tag{5.3}$$

Note: In accordance with our convention, the expression $\hat{c}^2 + \hat{s}^2$ is evaluated by squaring (with exact arithmetic) the floating point numbers $\hat{c}$ and $\hat{s}$ and then adding (with exact arithmetic) the squares.

Proof. Since $f_1^2 \geq f_r^2 \geq f_2^2$, it follows from F4 that

$$\mathrm{fl}(f_1^2 - f_2^2) \geq \mathrm{fl}(f_r^2 - f_2^2) \geq 0.$$

By F3, we have

$$0 \leq \mathrm{fl}\left(\frac{f_r^2 - f_2^2}{f_1^2 - f_2^2}\right) \leq 1.$$

Hence, F5 yields

$$0 \le \hat{c} = \mathrm{fl}\left(\frac{f_r^2 - f_2^2}{f_1^2 - f_2^2}\right)^{1/2} \le 1, \tag{5.4}$$

which gives the inequality for $\hat{c}$ in (5.3).

We now apply F1 and F2 to the expression for $\hat{c}$:

$$\hat{c} = \mathrm{fl}\left(\frac{f_r^2 - f_2^2}{f_1^2 - f_2^2}\right)^{1/2} = \left(\frac{f_r^2 - f_2^2}{f_1^2 - f_2^2}\right)^{1/2} \left(\frac{(1+\epsilon_1)(1+\epsilon_3)}{(1+\epsilon_2)}\right)^{1/2} (1+\epsilon_4), \tag{5.5}$$

where $|\epsilon_i| \le u$ for $1 \le i \le 4$. The $(1+\epsilon_1)$ and $(1+\epsilon_2)$ factors are connected with the subtractions, the $(1+\epsilon_3)$ factor comes from the division of numerator by denominator, and the $(1+\epsilon_4)$ factor comes from the square root. Squaring (5.5) and utilizing the bound $|\epsilon_i| \le u$, $1 \le i \le 4$, yields the estimate for $\hat{c}^2$ in (5.3).

Finally, we apply F1 and F2 to $\hat{s}$:

$$\hat{s} = \mathrm{fl}\left(1 - \hat{c}^2\right)^{1/2} = \left((1 - \hat{c}^2(1+\epsilon_1))(1+\epsilon_2)\right)^{1/2} (1+\epsilon_3), \tag{5.6}$$

where $|\epsilon_i| \le u$ for $i = 1, 2, 3$. The $(1+\epsilon_1)$ factor reflects the error in the squaring of $\hat{c}$, the $(1+\epsilon_2)$ factor is due to the subtraction, and the $(1+\epsilon_3)$ factor is due to the square root. Squaring (5.6) and utilizing (5.4), we see that

$$\hat{s}^2 = 1 - \hat{c}^2 + O(4u), \tag{5.7}$$

which establishes (5.2). By F5, (5.4), and (5.6), we conclude that $0 \le \hat{s} \le 1$, completing the proof of (5.3). □

Using Lemma 5.1, we show that the floating point matrices $\widehat{\mathbf{G}}_1$ and $\widehat{\mathbf{G}}_2$ are nearly unitary. The estimate for $\widehat{\mathbf{G}}_2$ is based on the following computation of its entries:

$$\widehat{\mathbf{G}}_2^* = \mathrm{fl}(r_k \mathbf{U}), \quad \text{where } \mathbf{U} = \mathrm{fl}\begin{bmatrix} \delta_1^*\left(\frac{c}{|r_k|^2}\right) & \delta_2^*\left(\frac{s}{|r_k|^2}\right) \\ -\delta_2\left(\frac{s}{|r_k|^2}\right) & \delta_1\left(\frac{c}{|r_k|^2}\right) \end{bmatrix}. \tag{5.8}$$

Lemma 5.2. *If $f_1^2 \ge f_r^2 \ge f_2^2$, $f_1^2 > f_2^2$, and the floating point arithmetic satisfies IEEE standard 754, then we have*

$$\mathbf{U}^*\mathbf{U} = \frac{1}{|r_k|^2}\begin{bmatrix} 1 + O(17u) & 0 \\ 0 & 1 + O(17u) \end{bmatrix}. \tag{5.9}$$

Proof. Since the factors multiplying δ_1 and δ_2 in (5.8) are real, it follows that the floating point matrix $\mathbf{U}$ has the following structure:

$$\mathbf{U} = \begin{bmatrix} a^* & b^* \\ -b & a \end{bmatrix}, \quad a = \mathrm{fl}\left[\delta_1\left(\frac{c}{|r_k|^2}\right)\right], \quad b = \mathrm{fl}\left[\delta_2\left(\frac{s}{|r_k|^2}\right)\right]. \tag{5.10}$$

Hence, the off-diagonal elements of $\mathbf{U}^*\mathbf{U}$ vanish. Now consider the diagonal elements. Suppose that $\delta_1^* = x + y\mathbf{i}$, where x and y are the real and imaginary parts of δ^*. Observe that

$$\begin{aligned} a^* = \mathbf{U}_{11} &= \mathrm{fl}\left(\frac{\delta_1^* \hat{c}}{|r_k|^2}\right) = \mathrm{fl}\left(\frac{x\hat{c}}{|r_k|^2}\right) + \mathrm{fl}\left(\frac{y\hat{c}}{|r_k|^2}\right)\mathbf{i} \\ &= \left(\frac{x\hat{c}}{f_r^2}\right)(1+\epsilon_1)(1+\epsilon_2) + \left(\frac{y\hat{c}}{f_r^2}\right)(1+\epsilon_1)(1+\epsilon_2')\mathbf{i}, \end{aligned} \tag{5.11}$$

where the $(1+\epsilon_1)$ factor is due to the division of $\hat{c}$ by f_r^2, and the $(1+\epsilon_2)$ and $(1+\epsilon_2')$ factors are associated with the multiplication by x and by y. Here $|\epsilon_i| \leq u$, $i = 1, 2$, and $|\epsilon_2'| \leq u$. Taking the norm and squaring yields:

$$|\mathbf{U}_{11}|^2 = \left(\frac{(x^2+y^2)\hat{c}^2}{(f_r^2)^2}\right)(1+O(4u)) = \left(\frac{|\delta_1|^2\hat{c}^2}{(f_r^2)^2}\right)(1+O(4u)).$$

Using (5.1), we substitute $|r_k|^2(1+O(2u))$ for one f_r^2 factor in the denominator to obtain

$$|\mathbf{U}_{11}|^2 = \frac{1}{|r_k|^2}\left(\frac{|\delta_1|^2\hat{c}^2}{f_r^2}\right)(1+O(6u)).$$

Again, using (5.1), we replace $|\delta_1|^2$ in the numerator by $f_1^2(1+O(2u))$ to obtain

$$|\mathbf{U}_{11}|^2 = \frac{1}{|r_k|^2}\left(\frac{f_1^2\hat{c}^2}{f_r^2}\right)(1+O(8u)).$$

In the same fashion,

$$|\mathbf{U}_{21}|^2 = \frac{1}{|r_k|^2}\left(\frac{f_2^2\hat{s}^2}{f_r^2}\right)(1+O(8u)).$$

Hence, we have

$$(\mathbf{U}^*\mathbf{U})_{11} = |\mathbf{U}_{11}|^2 + |\mathbf{U}_{21}|^2 = \frac{1}{|r_k|^2}\left(\frac{1}{f_r^2}\right)\left(f_1^2\hat{c}^2 + f_2^2\hat{s}^2\right)(1+O(8u)). \tag{5.12}$$

Let C^2 and S^2 be defined by

$$C^2 = \frac{f_r^2 - f_2^2}{f_1^2 - f_2^2} \quad \text{and} \quad S^2 = 1 - C^2 = \frac{f_1^2 - f_r^2}{f_1^2 - f_2^2}.$$

Observe that

$$f_1^2C^2 + f_2^2S^2 = f_r^2. \tag{5.13}$$

By Lemma 5.1, we can write

$$\hat{c}^2 = C^2 + e, \quad \text{where } e = O(5u)C^2 = O(5u)\left(\frac{f_r^2 - f_2^2}{f_1^2 - f_2^2}\right). \tag{5.14}$$

Also, by Lemma 5.1, we have

$$\hat{s}^2 = S^2 - e + O(4u).$$

These substitutions in (5.12) yield

$$\begin{aligned}
(\mathbf{U}^*\mathbf{U})_{11} &= \frac{1}{|r_k|^2}\left(\frac{1}{f_r^2}\right)\left(f_1^2\hat{c}^2 + f_2^2\hat{s}^2\right)(1+O(8u)) \\
&= \frac{1}{|r_k|^2}\left(\frac{1}{f_r^2}\right)\left(f_1^2(C^2+e) + f_2^2(S^2 - e + O(4u)\right)(1+O(8u)) \\
&= \frac{1}{|r_k|^2}\left(\frac{1}{f_r^2}\right)\left(f_1^2C^2 + f_2^2S^2 + (f_1^2 - f_2^2)e + f_2^2O(4u)\right)(1+O(8u)) \\
&= \frac{1}{|r_k|^2}\left[1 + \left(\frac{f_1^2 - f_2^2}{f_r^2}\right)e + O(4u)\right](1+O(8u)) \qquad (5.15) \\
&= \frac{1}{|r_k|^2}[1 + O(5u) + O(4u)](1+O(8u)) \qquad (5.16) \\
&= \left(\frac{1}{|r_k|^2}\right)(1+O(17u)).
\end{aligned}$$

To obtain (5.15), we utilize the identity (5.13) and the assumption $f_2^2 \le f_r^2$. To obtain (5.16), we also use the estimate (5.14) for e (established in Lemma 5.1). □

Theorem 5.3. *If $f_1^2 \ge f_r^2 \ge f_2^2$, $f_1^2 > f_2^2$, and the floating point arithmetic satisfies IEEE standard 754, then we have*

$$\widehat{\mathbf{G}}_1\widehat{\mathbf{G}}_1^* = \begin{bmatrix} 1+O(4u) & 0 \\ 0 & 1+O(4u) \end{bmatrix} \quad \textit{and} \tag{5.17}$$

$$\widehat{\mathbf{G}}_2\widehat{\mathbf{G}}_2^* = \begin{bmatrix} 1+O(23u) & O(6u) \\ O(6u) & 1+O(23u) \end{bmatrix}. \tag{5.18}$$

Proof. The identity (5.17) comes from (5.2). Now consider (5.18). Recall that r_k and δ_1 are in general complex. By Lemma 3.5 in [11], we have

$$\mathrm{fl}(r_k\delta_1^*) = r_k\delta_1^*(1+O(2\sqrt{2}u)).$$

Using the notation in (5.10), it follows that

$$\widehat{\mathbf{G}}_2^* = \mathrm{fl}(r_k\mathbf{U}) = \begin{bmatrix} (r_ka^*)(1+O(2\sqrt{2}u)) & r_kb^*(1+O(2\sqrt{2}u)) \\ -(r_kb)(1+O(2\sqrt{2}u)) & r_ka(1+O(2\sqrt{2}u)) \end{bmatrix}. \tag{5.19}$$

Hence,

$$\begin{aligned} (\widehat{\mathbf{G}}_2\widehat{\mathbf{G}}_2^*)_{11} &= |r_k|^2(|a|^2+|b|^2)(1+O(2\sqrt{2}u))^2 \\ &= |r_k|^2(|a|^2+|b|^2)(1+O(4\sqrt{2}u)). \end{aligned} \tag{5.20}$$

By Lemma 5.2, $|r_k|^2(|a|^2+|b|^2) = 1+O(17u)$. Making this substitution in (5.20) gives

$$\begin{aligned} (\widehat{\mathbf{G}}_2\widehat{\mathbf{G}}_2^*)_{11} &= (1+O(17u))(1+O(4\sqrt{2}u)) \\ &\le (1+O(17u))(1+O(6u)) = 1+O(23u). \end{aligned}$$

This establishes the expression in (5.18) for the the (1,1)-element. The (2,2)-element is similar.

The (2,1)-element in (5.18) can be expressed as

$$(\widehat{\mathbf{G}}_2\widehat{\mathbf{G}}_2^*)_{21} = |r_k|^2a^*b\left((1+O(2\sqrt{2}u))^2-(1+O(2\sqrt{2}u))^2\right).$$

Hence,

$$|(\widehat{\mathbf{G}}_2\widehat{\mathbf{G}}_2^*)_{21}| \le |r_k|^2|a||b|O(8\sqrt{2}u). \tag{5.21}$$

By Lemma 5.2, we have

$$|a||b| \le \frac{1}{2}(|a|^2+|b|^2) = \frac{1}{2}(\mathbf{U}^*\mathbf{U})_{11} = \left(\frac{1}{2|r_k|^2}\right)(1+O(17u)).$$

It follows from (5.21) that

$$|(\widehat{\mathbf{G}}_2\widehat{\mathbf{G}}_2^*)_{21}| \le (1+O(17u))O(4\sqrt{2}u) \le O(6u).$$

The (1,2)-element in (5.18) is similar. □

Theorem 5.3 does not imply that $\widehat{\mathbf{G}}_i$ is close to $\mathbf{G}_i$, $i=1,2$. It only states that when the $\mathbf{G}_i$ are evaluated using floating point arithmetic, the resulting floating point matrices are nearly unitary, even though the respective elements of $\widehat{\mathbf{G}}_i$ and $\mathbf{G}_i$ could differ by as much as one. Next, we show that when these nearly unitary matrices are used to evaluate the product (3.3), the elements on the diagonal and the subdiagonal of the product are close to their correct values. We do not analyze

the $(1,2)$ (superdiagonal) element in the product since its value is not important (and in fact, its value need not be close to the exact matrix element); what is important is that $\widehat{\mathbf{G}}_1$ and $\widehat{\mathbf{G}}_2$ are nearly unitary and the computed product is nearly upper triangular, with the diagonal elements and the subdiagonal element close to the exact elements.

Theorem 5.4. *If $f_1^2 \geq f_r^2 \geq f_2^2$, $f_1^2 > f_2^2$, and the floating point arithmetic satisfies IEEE standard* 754*, then with exact arithmetic, we have*

$$\widehat{\mathbf{G}}_2^* \boldsymbol{\Delta} \widehat{\mathbf{G}}_1 = \begin{bmatrix} r_k(1+O(16u)) & * \\ O(12u)|\delta_1| & y(1+O(11u)) \end{bmatrix}, \tag{5.22}$$

where

$$\boldsymbol{\Delta} = \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix},$$

and $y = \delta_1\delta_2 r_k/|r_k|^2$ is the exact $(2,2)$ element appearing in (3.5).

In each step of the GTD algorithm, we multiply a 2 by 2 diagonal matrix by the Givens rotations appearing in (3.3). With exact arithmetic, we should obtain an upper triangular matrix with r_k and y on the diagonal. According to Theorem 5.4, if the numerically evaluated Givens rotations are multiplied against the diagonal matrix, then with exact arithmetic we obtain almost the correct result. That is, the $(2,1)$ element differs from zero by a small multiple of u, and the diagonal elements are close, in a relative sense, to their correct values. Hence, if we simply put zero in the $(2,1)$ position and r_k and y on the diagonal, then we achieve nearly the same result that we would have gotten using exact arithmetic. The $(1,2)$ element in (5.22), shown as *, is not analyzed in the theorem since any error in it has no impact on the computation of the subsequent rotations. In each step of the GTD algorithm, the computation of the Givens rotations is expressed in terms of two diagonal elements in the partially triangularized matrix; as we show, the numerically evaluated rotations generate nearly the correct diagonal elements r_k and y of the triangular matrix.

Proof. Combining (5.11) and (5.19), we have

$$(\widehat{\mathbf{G}}_2^*)_{11} = \left(\frac{r_k \delta_1^* \hat{c}}{f_r^2}\right)(1+\epsilon_1)(1+\epsilon_2)(1+O(2\sqrt{2}u)),$$

where $|\epsilon_i| \leq u$, $i = 1, 2$. The (1,2)-element of $\widehat{\mathbf{G}}_2^*$ has the same form, but with c replaced by s and with δ_1 replaced by δ_2:

$$(\widehat{\mathbf{G}}_2^*)_{12} = \left(\frac{r_k \delta_2^* \hat{s}}{f_r^2}\right)(1+\epsilon_1')(1+\epsilon_2')(1+O(2\sqrt{2}u)),$$

where $|\epsilon_i'| \leq u$ for $i = 1, 2$. When the product $\widehat{\mathbf{G}}_2^* \boldsymbol{\Delta} \widehat{\mathbf{G}}_1$ is evaluated with exact arithmetic, we obtain

$$\begin{aligned}(\widehat{\mathbf{G}}_2^* \boldsymbol{\Delta} \widehat{\mathbf{G}}_1)_{11} &= \left(\frac{r_k|\delta_1|^2 \hat{c}^2}{f_r^2}\right)(1+O(2\sqrt{2}u))(1+\epsilon_1)(1+\epsilon_2) \\ &\quad + \left(\frac{r_k|\delta_2|^2 \hat{s}^2}{f_r^2}\right)(1+O(2\sqrt{2}u))(1+\epsilon_1')(1+\epsilon_2').\end{aligned}$$

Hence, we have

$$(\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1)_{11} = r_k\left(\frac{\hat{c}^2|\delta_1|^2+\hat{s}^2|\delta_2|^2}{f_r^2}\right)(1+O(5u)).$$

We replace $|\delta_1|^2$ and $|\delta_2|^2$ by $f_1^2(1+O(2u))$ and $f_2^2(1+O(2u))$ respectively, using (5.1), to obtain

$$\begin{aligned}(\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1)_{11} &= r_k\left(\frac{\hat{c}^2f_1^2+\hat{s}^2f_2^2}{f_r^2}\right)(1+O(2u))(1+O(5u))\\ &= r_k\left(\frac{\hat{c}^2f_1^2+\hat{s}^2f_2^2}{f_r^2}\right)(1+O(7u)). \qquad (5.23)\end{aligned}$$

In the proof of Lemma 5.2, in equation (5.16), we show that

$$\frac{\hat{c}^2f_1^2+\hat{s}^2f_2^2}{f_r^2} = 1+O(9u).$$

Combining this with (5.23) gives

$$(\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1)_{11} = r_k(1+O(9u))(1+O(7u)) = r_k(1+O(16u)),$$

the $(1,1)$-element in (5.22).

The (2,1)-element can be expressed

$$\begin{aligned}(\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1)_{21} &= \left(\frac{r_k\hat{s}\hat{c}\delta_1\delta_2}{f_r^2}\right)(1+O(2\sqrt{2}u))(1+\epsilon_1)(1+\epsilon_2)\\ &\quad-\left(\frac{r_k\hat{s}\hat{c}\delta_1\delta_2}{f_r^2}\right)(1+O(2\sqrt{2}u))(1+\epsilon_1')(1+\epsilon_2').\end{aligned}$$

Taking absolute values, we obtain

$$|(\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1)_{21}| = \left(\frac{\hat{s}\hat{c}|r_k||\delta_1||\delta_2|}{f_r^2}\right)O(10u).$$

Using (5.1), we replace $\sqrt{f_r^2}$ by $|r_k|\sqrt{(1+O(2u))}$ and we replace $|\delta_2|$ by $\sqrt{f_2^2}\sqrt{1+O(2u)}$. Since $0\le\hat{s},\hat{c}\le 1$ (see Lemma 5.1), we have

$$|(\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1)_{21}| = \left(\frac{|\delta_1|\sqrt{f_2^2}\sqrt{1+O(2u)}}{\sqrt{f_r^2}\sqrt{1+O(2u)}}\right)O(10u).$$

Since $f_2^2\le f_r^2$ and $\sqrt{1+O(2u)} = 1+O(u)$, it follows that

$$|(\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1)_{21}| = \left(\frac{|\delta_1|\sqrt{f_2^2}(1+O(u))}{\sqrt{f_r^2}(1+O(u))}\right)O(10u) = |\delta_1|O(12u).$$

In a similar fashion, the $(2,2)$ element can be expressed as

$$(\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1)_{22} = \left(\frac{r_k\delta_1\delta_2}{f_r^2}\right)(\hat{s}^2+\hat{c}^2)(1+O(5u)).$$

Substituting for $\hat{s}^2+\hat{c}^2$ using (5.2) and substituting for f_r^2 using (5.1), we obtain

$$\begin{aligned}(\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1)_{22} &= \left(\frac{r_k\delta_1\delta_2}{|r_k|^2(1+O(2u))}\right)(1+O(4u))(1+O(5u))\\ &= \left(\frac{r_k\delta_1\delta_2}{|r_k|^2}\right)(1+O(11u)).\end{aligned}$$

□

A MATLAB code implementing the GTD is posted at the following web site:

http://www.math.ufl.edu/∼hager/papers/GTD.

In our implementation of the GTD algorithm, we do not use floating point arithmetic to evaluate the product $\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1$, rather we insert r_k and y on the diagonal of the product, and 0 on the subdiagonal. Theorem 5.4 shows that if we compute the product $\widehat{\mathbf{G}}_2^*\boldsymbol{\Delta}\widehat{\mathbf{G}}_1$ with exact arithmetic, then the diagonal and subdiagonal elements are close to r_k, y, and 0.

In our analysis of the key step (3.3) in the GTD algorithm, it was assumed that $f_1^2 \geq f_r^2 \geq f_2^2$. On the other hand, due to the error terms in (5.22), there may not exist an index p satisfying (3.1) or there may not exist an index q satisfying (3.2). In the MATLAB code, we handle these cases in the following ways:

- If we cannot find an index p satisfying (3.1), then we set

$$p = \arg \max_i \{|\hat{r}_i^{(k)}| : k \leq i \leq K\}. \tag{5.24}$$

- If we cannot find an index q satisfying (3.2), then we set

$$p = \arg \min_i \{|\hat{r}_i^{(k)}| : k \leq i \leq K\}. \tag{5.25}$$

In either case, the following exchanges are performed:

$$\begin{aligned} R_{kk} &\leftrightarrow R_{pp}, \\ \mathbf{R}_{1:k-1,k} &\leftrightarrow \mathbf{R}_{1:k-1,p}, \\ \mathbf{P}_{:,k} &\leftrightarrow \mathbf{P}_{:,p}, \\ \mathbf{Q}_{:,k} &\leftrightarrow \mathbf{Q}_{:,p}. \end{aligned}$$

We choose $\mathbf{G}_1 = \mathbf{I}$, while $\mathbf{G}_2^*$ is the identity matrix with the k-th diagonal element replaced by $(r_k/|r_k|)(\delta_1^*/|\delta_1|)$.

The motivation for these choices is the following: If the index p in (3.1) does not exist, then the maximum in (5.24) must be very close to $|r_k|$. A symmetric permutation is performed to move the absolute largest diagonal element to the (k,k) position. The k-th diagonal element of $\mathbf{G}_2^*$ is chosen to have unit magnitude; its complex argument is chosen so that its product with δ_1 is a positive multiple of r_k, the desired k-th diagonal element of $\mathbf{R}$. When the index q in (3.2) does not exist, then the minimum in (5.25) must be very close to r_k. The choice of $\mathbf{G}_1$ and $\mathbf{G}_2$ is the same as before.

6. Inverse eigenvalue problems

In [4] Chu presents a recursive procedure for constructing matrices with prescribed eigenvalues and singular values. His algorithm, which he calls SVD_EIG, is based on Horn's divide and conquer proof of the sufficiency of Weyl's product inequalities. In general, the output of SVD_EIG is not upper triangular. Consequently, this routine could not be used to generate the GTD. Chu notes that to achieve an upper triangular matrix would require an algorithm "one order more expensive than the divide-and-conquer algorithm."

Given a vector of singular values $\boldsymbol{\sigma} \in \mathbb{R}^n$ and a vector of eigenvalues $\boldsymbol{\lambda} \in \mathbb{C}^n$, with $\boldsymbol{\lambda} \preceq \boldsymbol{\sigma}$, we can use the GTD to generate a matrix $\mathbf{R}$ with $\boldsymbol{\lambda}$ on the diagonal and with singular values $\boldsymbol{\sigma}$. In this section, we compare the solution to the inverse eigenvalue problem provided by the GTD to Chu's algorithm. In our initial experimentation,

TABLE 6.1. Comparison of SVD_EIG and GTD for inverse eigenvalue problems (CPU time in seconds, relative errors in singular values and eigenvalues in sup-norm)

	Time		σ error		λ error	
Dimension	SVD_EIG	GTD	SVD_EIG	GTD	SVD_EIG	GTD
100	0.61	0.20	1.9e−16	2.0e−16	6.6e−16	0
200	2.24	0.38	2.0e−16	1.7e−16	5.9e−15	0
400	13.84	0.86	3.4e−16	1.8e−16	1.7e−15	0
800	97.50	2.30	2.5e−16	1.8e−16	3.7e−13	0
1200	317.83	5.67	1.8e−16	2.1e−16	2.5e−12	0
1600	746.77	10.77	4.0e−16	1.8e−16	9.6e−7	0

we discovered that the algorithm of Chu, as presented in [4], did not work. When this was pointed out, Chu provided an adjustment in which the parameter μ in [4, (2.2)] was replaced by $\mu\lambda_1/|\lambda_1|$. With this adjustment, it was possible to solve 4 by 4 and 5 by 5 test cases that previously caused failure. The results reported in this section use the adjusted algorithm.

Both MATLAB routines and SVD_EIG [4] require $O(n^2)$ flops, so in an asymptotic sense, the approaches are equivalent. In Table 6.1 we compare the actual running times of GTD and SVD_EIG for matrices of various dimensions. These computer runs were performed on a Sun Workstation with 2 GB memory. In making these runs, the portion of the GTD code connected with the updating of the matrices $\mathbf{P}$ and $\mathbf{Q}$ was deleted since SVD_EIG does not accumulate the unitary matrices. The input arrays $\boldsymbol{\sigma}$ and $\boldsymbol{\lambda}$ were generated in the following way: Using the MATLAB routine RAND, we randomly generated a square matrix whose element lies between 0 and 1. The singular values $\boldsymbol{\sigma}$ were computed using the MATLAB routine SVD, and the eigenvalues $\boldsymbol{\lambda}$ were computed using MATLAB's EIG. By the theorem of Weyl [24], $\boldsymbol{\lambda} \preceq \boldsymbol{\sigma}$. We then used both SVD_EIG and GTD to generate matrices with the specified singular values and eigenvalues. Five different matrices of each dimension were generated, and the average running time is reported in Table 6.1.

The times shown in Table 6.1 indicate that GTD becomes increasingly more efficient than SVD_EIG as the matrix dimension increases. For a dimension of 100, GTD is about three times faster than SVD_EIG. For a dimension of 1600, GTD is about 70 times faster than SVD_EIG.

In Table 6.1 we also compare the specified singular values and eigenvalues to those obtained by applying MATLAB's SVD and EIG routines to the generated matrices. That is, for each matrix output by either SVD_EIG or GTD, we use MATLAB's routines to compute the singular values and eigenvalues. The relative difference between the singular values and eigenvalues generated by MATLAB's routines and the specified singular values and eigenvalues is evaluated in the sup-norm. The errors reported in Table 6.1 are the average errors for the 5 random matrices of each dimension. Both routines generate matrices with singular values that match those computed by MATLAB's SVD routine to within 16 digits. Observe that GTD always matches exactly the prescribed eigenvalues since the generated matrix is triangular, with the specified eigenvalues on the diagonal. The error in the eigenvalues of the matrix generated by SVD_EIG was comparable to the singular value error for matrices of dimension up to 400. Thereafter, the error in the eigenvalues grew

quickly. When the matrix dimension doubled from 400 to 800, the error increased roughly by the factor 10^2. Also, when the matrix dimension doubled again from 800 to 1600, the error increased roughly by the factor 10^5.

A recursive algorithm can require a significant amount of memory. While SVD_EIG executed, we monitored the memory usage with the Unix "top" command. We observed that for a matrix of dimension 1600, the memory consumption grew to 319 MB. Since a complex double precision matrix of dimension 1600 occupies about 41 MB memory, the recursion required more than 7 times as much space as the matrix itself.

7. Conclusions

By the theorem of Weyl [24], the generalized triangular decomposition represents the most general unitary decomposition $\mathbf{H} = \mathbf{QRP}^*$. That is, the diagonal $\mathbf{r}$ of $\mathbf{R}$ must satisfy $\mathbf{r} \preceq \boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ is the vector of singular values for $\mathbf{H}$, while for any diagonal $\mathbf{r}$ with $\mathbf{r} \preceq \boldsymbol{\sigma}$, we can write $\mathbf{H} = \mathbf{QRP}^*$. The GTD includes, as special cases, the singular value decomposition, the Schur decomposition, the $\mathbf{QR}$ decomposition, and the geometric mean decomposition. Given the SVD, the GTD based on $\mathbf{r}$ can be evaluated using a series of Givens rotations and permutations. The GTD algorithm provides a new proof of Horn's theorem [12]. Applications of the GTD include transceiver design for MIMO communications [16, 17, 18] and inverse eigenvalue problems, surveyed extensively in [3]. In terms of CPU time and memory requirements, GTD is superior to a recursive approach for generating matrices with specified singular values and eigenvalues. The GTD update step is backed by a rigorous numerical stability theory developed in Section 5. The numerical results reported in Section 6 are an indication that the overall algorithm has strong stability properties.

References

[1] *IEEE Standard for Binary Floating-Point Arithmetic ANSI/IEEE Standard 754-1985*, Institute of Electrical and Electronics Engineers, New York, 1985.

[2] E. Beltrami, *Sulle funzioni bilineari*, Giornale De Matematiche, 11 (1873), pp. 98–106.

[3] M. T. Chu, *Inverse eigenvalue problems*, SIAM Rev., 40 (1998), pp. 1–39 (electronic). MR1612561 (99e:15008)

[4] M. T. Chu, *A fast recursive algorithm for constructing matrices with prescribed eigenvalues and singular values*, SIAM J. Numer. Anal., 37 (2000), pp. 1004–1020. MR1749246 (2001d:65044)

[5] W. Givens, *Computation of plane unitary rotations transforming a general matrix to triangular form*, SIAM J. Appl. Math., 6 (1958), pp. 26–50. MR0092223 (19:1081e)

[6] G. H. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM Journal on Numerical Analysis, 2 (1965), pp. 205–224. MR0183105 (32:587)

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983. MR733103 (85h:65063)

[8] T. Guess, *Optimal sequence for CDMA with decision-feedback receivers*, IEEE Trans. Inform. Theory., 49 (2003), pp. 886–900.

[9] W. W. Hager, *Applied Numerical Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[10] R. J. Hanson and C. L. Lawson, *Extensions and applications of the Householder algorithm for solving linear least square problems*, Math. Comp., 23 (1969), pp. 787–812. MR0258258 (41:2905)

[11] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996. MR1368629 (97a:65047)

[12] A. Horn, *On the eigenvalues of a matrix with prescribed singular values*, Proc. Amer. Math. Soc., 5 (1954), pp. 4–7. MR0061573 (15:847d)

[13] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991. MR1091716 (92e:15003)
[14] A. S. Householder, *Unitary triangularization of a nonsymmetric matrix*, ACM, 5 (1958), pp. 339–342. MR0111128 (22:1992)
[15] Y. Jiang, W. W. Hager, and J. Li, *The geometric mean decomposition*, Linear Algebra Appl., 396 (2005), pp. 373–384. MR2112215 (2005h:15045)
[16] ———, *Tunable channel decomposition for MIMO communications using channel state information*, IEEE Trans. Signal Process., 54 (2006), pp. 4405–4418.
[17] Y. Jiang, J. Li, and W. W. Hager, *Joint transceiver design for MIMO communications using geometric mean decomposition*, IEEE Trans. Signal Process., 53 (2005), pp. 3791–3803. MR2239898
[18] ———, *Uniform channel decomposition for MIMO communications*, IEEE Trans. Signal Process., 53 (2005), pp. 4283–4294. MR2242173
[19] C. Jordan, *Mémoire sur les formes bilinéaires*, J. Math. Pures Appl., 19 (1874), pp. 35–54.
[20] P. Kosowski and A. Smoktunowicz, *On constructing unit triangular matrices with prescribed singular values*, Computing, 64 (2000), pp. 279–285. MR1767057 (2001e:65073)
[21] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980. MR570116 (81j:65063)
[22] I. Schur, *On the characteristic roots of a linear substitution with an application to the theory of integral equations*, Math. Ann., 66 (1909), pp. 488–510.
[23] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997. MR1444820 (98k:65002)
[24] H. Weyl, *Inequalities between two kinds of eigenvalues of a linear transformation*, Proc. Nat. Acad. Sci. U. S. A., 35 (1949), pp. 408–411. MR0030693 (11:37d)
[25] J. H. Wilkinson and C. Reinsch, *Linear algebra*, in Handbook for Automatic Computation, F. L. Bauer, ed., vol. 2, Berlin, 1971, Springer-Verlag. MR0461856 (57:1840)
[26] J.-K. Zhang, T. N. Davidson, and K. M. Wong, *Uniform decomposition and mutual information using MMSE decision feedback detection*, in Proceedings of International Symposium on Information Theory, 2005, pp. 714–718.
[27] J.-K. Zhang, A. Kavčić, and K. M. Wong, *Equal-diagonal QR decomposition and its application to precode design for successive-cancellation detection*, IEEE Trans. Inform. Theory., 51 (2005), pp. 154–172. MR2234579

Department of Electrical and Computer Engineering, University of Florida, P.O. Box 116130, Gainesville, Florida 32611-6130

Current address: Department of Electrical and Computer Engineering, University of Colorado, Boulder, Colorado 80309-0425

E-mail address: yjiang@dsp.ufl.edu

Department of Mathematics, University of Florida, P.O. Box 118105, Gainesville, Florida 32611-8105

E-mail address: hager@math.ufl.edu

URL: http://www.math.ufl.edu/~hager

Department of Electrical and Computer Engineering, P.O. Box 116130, University of Florida, Gainesville, Florida 32611-6130

E-mail address: li@dsp.ufl.edu