Computing 64, 279–285 (2000)

Computing

On Constructing Unit Triangular Matrices with Prescribed Singular Values

P. Kosowski and **A. Smoktunowicz,** Warsaw

Received July 21, 1999; revised November 4, 1999

Abstract

We propose an efficient algorithm for computing a unit lower triangular $n \times n$ matrix with prescribed singular values of $O(n^2)$ cost. This is a solution of the question raised by N. J. Higham in [4, Problem 26.3, p. 528].

AMS Subject Classifications: 15A18, 65F30.

Key Words: Singular value decomposition, stability, triangular matrix.

1. Introduction

In [4] N. J. Higham raised the following research problem (Problem 26.3, p. 528):

> Develop an efficient algorithm for computing a unit upper triangular $n \times n$ matrix with prescribed singular values $\sigma_1, \ldots, \sigma_n$, where $\prod_i \sigma_i = 1$.

The main goal of this paper is to construct an algorithm for solving this problem, which can find many applications in applied numerical linear algebra ([9]).

Let $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0$ and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ denote the eigenvalues and the singular values of a given matrix $A \in \mathbb{R}^{n \times n}$. We remind that the singular values of A are the nonnegative square roots of the eigenvalues of A^*A, where A^* is the conjugate transpose of A. In 1949 Weyl showed that $|\lambda_1 \cdots \lambda_k| \leq \sigma_1 \cdots \sigma_k$ for $k = 1, \ldots, n$ and the equality holds when $k = n$ (the so-called Weyl conditions). Moreover, A. Horn [6] proved that the Weyl conditions are sufficient for the existence of triangular matrix with prescribed singular values and eigenvalues. Yet the another proof can be found in [7]. The idea of our algorithm is inspired by [6]. The numerical examples by means of the *MATLAB* function are also given.

A part of this work was presented as a poster of Alicja Smoktunowicz and Marek Aleksiejczyk displayed in the SIAM Applied Linear Algebra Conference in Snowbird, 1997.

2. Description of the Algorithm

We would like to find a unit lower triangular matrix $A \in \mathbb{R}^{n\times n}$ with prescribed singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$. Due to A. Horn's theorem ([6]) such matrix A exists if and only if the Weyl conditions are satisfied:

$$\forall_{i=2,\ldots,n}\ \sigma_1\sigma_2\cdots\sigma_{i-1} \geq 1 \quad \text{and} \quad \sigma_1\sigma_2\cdots\sigma_n = 1. \tag{1}$$

The singular values of $A \in \mathbb{R}^{n\times n}$ are the positive square roots of the eigenvalues of the symmetric matrix A^TA and can be found from the Singular Value Decomposition SVD ([2], [3]).

Theorem 1 (Singular Value Decomposition). *If $A \in \mathbb{R}^{n\times n}$ has rank* r*, then there exist orthogonal matrices $U, V \in \mathbb{R}^{n\times n}$ such that $A = U\Sigma V^T$, where $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ (the so-called singular values of A).*

The idea of our algorithm is based on the following two lemmas. The main transformation in this algorithm is reduced to finding the SVD of a lower triangular 2×2 matrix.

Lemma 2. *For every real $a, b > 0$ such that $a \geq 1 \geq b$ or $b \geq 1 \geq a$ the matrix $L \in \mathbb{R}^{2\times 2}$ of the form*

$$L(a,b) = \begin{pmatrix} 1 & 0 \\ \sqrt{(a^2-1)(1-b^2)} & ab \end{pmatrix} \tag{2}$$

has singular values a and b.

Lemma 3. *Let real numbers $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ satisfy the Weyl conditions (1). Then there exists a permutation $\{d_1, d_2, \ldots, d_n\}$ of the set $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ such that the following inequalities hold*

$$\forall_{i=2,\ldots,n} \begin{cases} d_1d_2\cdots d_{i-1} \geq 1 \geq d_i \\ or \\ d_i \geq 1 \geq d_1d_2\cdots d_{i-1}, \end{cases} \tag{3}$$

and $d_1d_2\cdots d_n = 1$.

As the proof of Lemma 2 is easily seen, we would like to focus on the scheme of choosing such permutation in our algorithm. It is obvious that one can find many permutations of given set of singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ satisfying (3). This process sometimes involves multiplications of very large and very small numbers, so firstly we form an auxiliary vector $\tilde{\sigma}$ by sorting the σ_i ascendently. Without loss of generality we may assume that

$$\tilde{\sigma}_1 \leq \tilde{\sigma}_2 \leq \cdots \leq \tilde{\sigma}_l < 1 \leq \tilde{\sigma}_{l+1} \leq \cdots \leq \tilde{\sigma}_n.$$

Then we start forming new vector $d = [d_1, d_2, \ldots, d_n]^T$ from the middle of $\tilde{\sigma} = [\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_n]^T$; precisely by d_1 we take $\tilde{\sigma}_l$ and $d_2 = \tilde{\sigma}_{l+1}$. Further, if $d_1 d_2 < 1$ then $d_3 = \tilde{\sigma}_{l+2}$, and in the opposite cases $d_3 = \tilde{\sigma}_{l-1}$; and we proceed similarly for the product $d_1 d_2 \cdots d_{i-1}$ in (3).

The idea of our algorithm is to construct a sequence of unitarily equivalent lower triangular matrices $A_i \in \mathbb{R}^{n\times n}, i = 1, \ldots, n$ with diagonal matrix $\operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$, where $\sigma_1, \sigma_2, \ldots, \sigma_n$ are given singular values. We remind that two matrices M, N are said to be unitarily equivalent (notation: $M \sim N$) if there exit unitary matrices U, V such that $M = UNV$. Notice that the singular values are invariant under unitary equivalences.

Algorithm. The following algorithm computes a unit lower triangular matrix $A \in \mathbb{R}^{n\times n}$ for given set of singular values $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, which satisfy (1).

I. Permute the set $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, to obtain the vector d, whose elements $d_1, d_2, \ldots, d_n$ fulfill (3).

II. Using the method described below, find a unit triangular matrix $A \in \mathbb{R}^{n\times n}$, such that $A \sim \operatorname{diag}(d_1, d_2, \ldots, d_n)$.

We construct a sequence of unitarily equivalent lower triangular matrices $A_i \in \mathbb{R}^{n\times n}, i = 1, \ldots, n$ with diagonal matrix $\operatorname{diag}(d_1, d_2, \ldots, d_n)$.

Step 1. Let $A_1 = \operatorname{diag}(d_1, d_2, \ldots, d_n)$.

Step 2. Set

$$A_2 = \begin{pmatrix} L_2 & 0 \\ 0 & \operatorname{diag}(d_3, \ldots, d_n) \end{pmatrix},$$

where L_2 is 2×2 real matrix of the form (2), i.e. $L_2 = L(d_1, d_2)$.

Step 3. For each $i = 3, \ldots, n$ and for given lower triangular matrix A_{i-1} we will construct A_i. Suppose that A_{i-1} has the following form

$$A_{i-1} = \begin{pmatrix} B_{i-1} & 0 & 0 \\ C_{i-1} & \operatorname{diag}(a, b) & 0 \\ 0 & 0 & D_{i-1} \end{pmatrix},$$

where B_{i-1} is a unit lower triangular $(i-2) \times (i-2)$ matrix, $C_{i-1} = \begin{pmatrix} * & * & \cdots & * \\ 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{2\times(i-2)}, a = d_1, d_2 \cdots d_{i-1}, b = d_i$ and $D_{i-1} = diag(d_{i+1}, \ldots, d_n)$ (according to the $MATLAB$ notation we have $D_{n-1} = D_n = [])$.

Let $L_i = L(a, b)$ and decompose L_i such that

$$L_i = U_i \operatorname{diag}(a, b) V_i^T, \tag{4}$$

where $U_i, V_i \in \mathbb{R}^{2\times 2}$ are orthogonal. Now we are in position to construct $A_i \sim A_{i-1}$ such that $A_i = Q_i A_{i-1} Z_i^T$, where $Q_i, Z_i \in \mathbb{R}^{n\times n}$ are orthogonal and have the form

$$Q_i = \begin{pmatrix} I_{i-2} & 0 & 0 \\ 0 & U_i & 0 \\ 0 & 0 & I_{n-i} \end{pmatrix}, \quad Z_i = \begin{pmatrix} I_{i-2} & 0 & 0 \\ 0 & V_i & 0 \\ 0 & 0 & I_{n-i} \end{pmatrix},$$

where I_k denotes the k-th identity matrix. Then the matrix A_i has the structure

$$A_i = \begin{pmatrix} B_{i-1} & 0 & 0 \\ U_i C_{i-1} & L_i & 0 \\ 0 & 0 & D_{i-1} \end{pmatrix}, \tag{5}$$

so the only thing to compute is the product $U_i C_{i-1} \in \mathbb{R}^{2\times(i-2)}$.

Step 4. Set $A = A_n$.

Remark 1. In order to obtain a decomposition of a lower triangular 2×2 matrix L_i given in (4), i.e. $L_i = U_i \operatorname{diag}(a,b) V_i^T$, where U_i, V_i are 2×2 orthogonal matrices we can use a stable SVD decomposition (for example in *MATLAB*). If $a < b$ and we have the SVD decomposition $L_i = \tilde{U}_i \operatorname{diag}(b,a) \tilde{V}_i^T$, where $\tilde{U}_i, \tilde{V}_i$ are orthogonal, then we set $U_i = \tilde{U}_i T, T$ is a permutation matrix $T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Other stable algorithms for computing the SVD decomposition of a triangular 2×2 matrix are outlined in [2, Sec. 2.6.6]. Notice also that U_i, V_i are either Householder or Givens matrices, i.e. are in one of the following two forms $\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix}$.

At the end of this section we present the code of our algorithm in *M*-file in *MATLAB*.

```
function A=ut_svd(d);
%   UT_SVD Unit triangular matrix with prescribed singular
    values.
%   A=ut_svd(d) produces unit triangular matrix A (n×n)
%   with prescribed singular values
%   d(1)>=d(2)>=···>=d(n-1)>=d(n) such that
%   d(1) >= 1, d(1) * d(2) >= 1, ..., d(1) * d(2) * ··· d(n - 1) >= 1
%   and d(1) * d(2) * d(n) = 1 (the Weyl conditions).
%   References:
%   N. J. Higham, Accuracy and Stability of Numerical
%   Algorithms, SIAM, Philadelphia, 1996, Research Problem
%   No. 26.3, p. 528.
```

```
%   A. Horn, On the eigenvalues of a matrix with prescribeda
%   singular values, Proc. Amer. Math. Soc. 5, 1954, pp. 4--7.
n = max(size (d));
d = d(:)';  % Ensure d is a row vector.
dd = sort(d);
% Permute of d(1), d(2), ..., d(n).
k = find(dd < 1);
if == [], A = eye(n); return; end
ℓ = max(k);
a = dd(ℓ);
b = dd(ℓ + 1);
d = [a, b];
dd(ℓ) = 0;
dd(ℓ + 1) = 0;
i = 1; j = 0;
while any(dd)
  c = a * b;
  if c < 1,
    i = i + 1;
    y = dd(l + i);
    p = ℓ+i;
  else
    j = j + 1;
    y = dd(ℓ − j);
    p = ℓ − j;
  end
  d = [d, y];
  dd(p) = 0;
  a = c; b=y;
end
% Construct matrix A.
A = diag(d);
a = d(1); b = d(2);
pom = ((a − 1) * (1 − b)) * ((a + 1) * (b + 1));
c = sqrt(pom);
L = [1 0; c a * b];
A(1 : 2, 1 : 2) = L;
il = cumprod (d);
for s = 3 : n,
  a = il(s − 1);
  b = d(s);
  pom = ((a − 1) * (1 − b)) * ((a + 1) * (b + 1));
  c = sqrt(pom);
  L = [1 0; c a * b];
  [U, S, V] = svd(L);
```

```
  if a < b, U = U(:,[2,1]); end %Interchange columns of U (2 × 2).
  A(s - 1 : s, 1 : (s - 2)) = U * A(s - 1 : s, 1 : (s - 2));
  A(s - 1 : s, s - 1 : s) = L;
end
```

3. Numerical Examples

To illustrate our results we present some numerical experiments. All computations were carried out in *MATLAB* with unit roundoff $\rho = 2.2e - 16$. We used the SVD decomposition ([2]) to compute the singular values.

Notice that the cost of the algorithm is $3n^2$ flops plus $(n - 1)$ calls of SVD for a 2×2 matrix for the stage II, where a flop is a floating operation [4]. Thus the total cost of our algorithm is of order $O(n^2)$, counting the sorting of an input vector d, which may satisfy the Weyl conditions but not be ordered descendently (the expected complexity is $1.4n \log n$).

Numerical properties of the proposed algorithm depend mainly on the accuracy of SVD routine for 2×2 triangular matrix. A very sophisticated algorithm SLASV2 for computing singular values and vectors of triangular 2×2 matrix to nearly full machine precision is given in [1]. Our algorithm is done by applying stable 2×2 orthogonal transformations, hence possesses good numerical efficiency.

In each example we evaluate relative error between the given vector $d = [d_1, \ldots, d_n]^T$ of prescribed singular values and vector $s = \mathrm{svd}(H)$, where H is the lower triangular matrix obtained from our algorithm. We remind the well known the Wielandt-Hoffman theorem [2] on the sensitivity of singular values.

Theorem 4 (Wielandt-Hoffman). *Let A and $\tilde{A} = A + E \in \mathbb{R}^{n \times n}$ have singular values $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_n$ and $\tilde{\sigma}_1 \geq \tilde{\sigma}_2 \cdots \geq \tilde{\sigma}_n$, respectively. Then*

$$\|\sigma - \tilde{\sigma}\|_2 \leq \|E\|_F, \tag{6}$$

where $\sigma = [\sigma_1, \ldots, \sigma_n]^T$ and $\tilde{\sigma} = [\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n]^T$.

Example 5. The first example is the problem with a vector d of singular values derived from the Pascal matrix $A(n \times n)$, i.e. $A = \mathrm{pascal}\ (n)$. For given A we form the lower triangular matrix $L(L = \mathrm{tril}(A))$ and then compute a unit triangular one as follows $B = \mathrm{diag}(a_{11}^{-1}, \ldots, a_{nn}^{-1}) * L$. Next as a vector d we take $d = \mathrm{svd}(B)$ and execute our *MATLAB* script: $H = ut\_svd(d)$. The singular values of matrix B are very close to each other, so this is another challenge for our method. For different n the relative errors $\mathrm{err} = \frac{\|d - s\|_2}{\|d\|_2}, s = \mathrm{svd}(H)$ are shown in the table given below:

n	5	10	20	50	70
err	$1.35e{-}16$	$4.00e{-}16$	$3.95e{-}16$	$7.75e{-}16$	$7.95e{-}16$

It is worth noting that in each cases relative error err has the same magnitude as the unit roundoff.

Example 6. For the purpose of this example we have written *a* M-file `weyl (n,k)`[1], which randomly generates vectors d satisfying the Weyl conditions, where n denotes the length of d and k is the number of entries d such that $d_i > 1$. As in the previous example we present the results for various vectors d of prescribed singular values and obtained relative errors in the table given below.

d	weyl (7, 3)	weyl (15, 7)	weyl (50, 15)	weyl (70, 40)	weyl (100, 20)
err	$3.63e{-16}$	$6.65e{-16}$	$8.46e{-16}$	$6.68e{-16}$	$5.44e{-16}$

The efficiency of our algorithm, as the relative error is concerned, is again worth mentioning.

References

[1] Bai, Z., Demmel, J. W.: Computing the generalized singular value decomposition. SIAM J. Sci. Comput. *14*, 1464–1486 (1993).
[2] Björck, Å.: Numerical methods for least squares problems. SIAM: Philadelphia, 1996.
[3] Golub, G. H., Van Loan, C. F.: Matrix computations, 3rd ed. The Johns Hopkins University Press: Baltimore, 1996.
[4] Higham, N. J.: Accuracy and stability of numerical algorithms. SIAM: Philadelphia, 1996.
[5] Higham, N. J.: Notes on accuracy and stability of algorithms in numerical linear algebra. In: Proceedings of EPSRC Numerical Analysis Summer School, Leicester University, July 1988 (to appear).
[6] Horn, A.: On the eigenvalues of a matrix with prescribed singular values. Proc. Amer. Math. Soc. *5*, 4–7 (1954).
[7] Horn, R. A., Johnson, C. R.: Topics in matrix analysis. Cambridge: Cambridge University Press, 1991.
[8] Li, C.-K., Mathias, R.: An efficient algorithm for constructing matrices with prescribed singular values and eigenvalues. Manuscript, December 1997.
[9] Kosowski, P., Smoktunowicz, A.: Computing a unit triangular matrix with prescribed singular values, Preprint Nr 70, Politechnika Warszawska, Instytut Matematyki, December 1997.

Przemysław Kosowski
Institute of Mathematics
Polish Academy of Sciences
Sniadeckich 8, P.O. Box 137
00–950 Warsaw
Poland
e-mail: kswsk@impan.gov.pl

Alicja Smoktunowicz
Institute of Mathematics
Warsaw University of Technology
Pl. Politechniki 1
00–661 Warsaw
Poland
e-mail: smok@im.pw.edu.pl

[1] The source of `weyl (n,k)` is also available from the authors.