Electronic Transactions on Numerical Analysis.
Volume 44, pp. 593–623, 2015.

ISSN 1068–9613.

ETNA
Kent State University
http://etna.math.kent.edu

SVD OF HANKEL MATRICES IN VANDERMONDE-CAUCHY PRODUCT FORM*

ZLATKO DRMAČ†

Abstract. Structured matrices of Cauchy, Vandermonde, Hankel, Toeplitz, and other types arise in a variety of applications, and their SVD decomposition provides key information, e.g., in various rational approximation tasks. In particular, Hankel matrices play an important role in the Adamyan-Arov-Krein and Carathéodory-Feyér rational approximation theories as well as in various applications in signal processing and control theory. This paper proposes new algorithms to compute the SVD of a Hankel matrix given implicitly as the product $\mathcal{V}^T D\mathcal{V}$, where $\mathcal{V}$ is a complex Vandermonde matrix and D is a diagonal matrix. The key steps are the discrete Fourier transform and the computation of the SVD of $\mathcal{C}^T \widetilde{D}\mathcal{C}$, where $\mathcal{C}$ is a Cauchy matrix and $\widetilde{D}$ is diagonal. This SVD is computed by a specially tailored version of the Jacobi SVD for products of matrices. Error and perturbation analysis and numerical experiments confirm the robustness of the proposed algorithms, capable of computing to high relative accuracy all singular values in the full range of machine numbers.

Key words. Cauchy matrix, discrete Fourier transform, eigenvalues, Hankel matrix, Jacobi method, rational approximations, singular value decomposition, Toeplitz matrix, Vandermonde matrix

AMS subject classifications. 15A09, 15A12, 15A18, 15A23, 65F15, 65F22, 65F35

1. Introduction.

Cauchy, Vandermonde, Toeplitz, and Hankel matrices with various generalizations are the key objects in many areas of numerical mathematics and computational and engineering sciences, e.g., in signal processing and in particular in the Adamyan-Arov-Krein and Carathéodory-Feyér rational approximation theories. In a host of approximation methods, the coefficients of the approximants are taken from the singular vectors corresponding to small singular values of certain matrices of these kinds; see e.g., [29, 32, 33]. The fact that these structured matrices are extremely ill-conditioned is often the main obstacle that precludes turning powerful theoretical results into practical numerical procedures.

In this paper, we study the possibility of computing to high accuracy the SVD of Hankel matrices $\mathcal{H}(h)_{ij} = h_{i+j-1}$, $h \in \mathbb{C}^{2n-1} \setminus \{0\}$. As an illustration of the ill-conditioning, it suffices to mention that, e.g., the spectral condition number $\kappa_2(\mathcal{H}) = \|\mathcal{H}\|_2\|\mathcal{H}^{-1}\|_2$ of a real positive definite $n \times n$ Hankel matrix $\mathcal{H}$ is bounded from below by $3 \cdot 2^{n-6}$ (such that, for instance, $\kappa_2(\mathcal{H}) > 7.5 \cdot 10^{58}$); see, e.g., [48]. This means that an accurate finite-precision (floating-point) computation of the SVD of a Hankel matrix by a standard norm-wise backward stable algorithms is numerically feasible at best only for small dimensions. Indeed, if an algorithm computes the singular values $\sigma_1 \geq \cdots \geq \sigma_n > 0$ of $\mathcal{H}$ as $\widetilde{\sigma}_1 \geq \cdots \geq \widetilde{\sigma}_n \geq 0$, then the computed values $\widetilde{\sigma}_i$ correspond to a backward perturbed matrix $\mathcal{H} + \Delta\mathcal{H}$ with $\|\Delta\mathcal{H}\|_2 \leq \epsilon\|\mathcal{H}\|_2$, where ϵ is, up to a moderate factor of the dimension, of the order of machine precision $\widehat{\varepsilon}$. In general, $\mathcal{H} + \Delta\mathcal{H}$ has no Hankel structure. Perturbation theory yields $|\widetilde{\sigma}_i - \sigma_i| \leq \epsilon\sigma_1$ for all i, and the relative error in each computed singular value is bounded by

$$\frac{|\widetilde{\sigma}_i - \sigma_i|}{\sigma_i} \leq \epsilon\frac{\sigma_1}{\sigma_i} \leq \epsilon\kappa_2(\mathcal{H}), \quad \text{where, e.g., } \epsilon \approx O(10^{-16}) \text{ and } \kappa_2(\mathcal{H}) > 10^{50}, n = 200.$$

Hence, for those singular values σ_i that are much smaller than σ_1, no accurate digit will be revealed. Also, in this setting, the singular vector corresponding to any, say, simple singular

*Received May 4, 2015. Accepted September 10, 2015. Published online on November 19, 2015. Recommended by D. Potts. This work is supported by the grant HRZZ–9345 (*Mathematical modeling, analysis and computing with applications to complex mechanical systems*) from the Croatian Science Foundation. Part of this work was done while the author was a visiting professor at the Department of Mathematics and a visiting scientis supported by the Interdisciplinary Center for Applied Mathematics (ICAM) at the Virginia Polytechnic Institute and State University, Blacksburg, VA 24060.

†Faculty of Science, Department of Mathematics, University of Zagreb, Bijenička 30, 10000 Zagreb, Croatia (drmac@math.hr).

ETNA
Kent State University
http://etna.math.kent.edu

value σ_i will be badly approximated, in particular, if the gap $\gamma(\sigma_i) \equiv \min_{\sigma_j \neq \sigma_i} |\sigma_j - \sigma_i|/\sigma_1$ is small. Is an accuracy better than this possible?

Unfortunately, an accurate SVD of $\mathcal{H}$ as a function of h (i.e., of $\mathcal{H}_{ij} = h_{i+j-1}$) is a mission impossible. Namely, computing the singular values of $\mathcal{H}$ to high accuracy for any input vector h (and thus computing its determinant to high accuracy for any h) is out of question because it is always impossible to compute the determinant $\det(\mathcal{H})$ accurately as a function of h. This is a corollary of the analogous claim proved for Toeplitz matrices [14, Section 2.6] simply because $\mathcal{T} = P\mathcal{H}$ is Toeplitz with a permutation matrix P with $\det(P) = (-1)^{n-1}$. The difficulties in computing $\det(\mathcal{T})$ have nontrivial origins, specifically for a complex Toeplitz matrix $\mathcal{T}$ by the irreducibility of $\det(\mathcal{T})$ over any field or for a real one because $\nabla\det(\mathcal{T})$ has all nonzero entries on a Zariski-open set. For a full theoretical analysis, we refer to the fundamental work of Demmel, Dumitriu, and Holtz [14].

However, in some cases, the Hankel matrix can be given implicitly parametrized in terms of a new set of variables. For instance, the Hilbert matrix is a Hankel matrix, but it is also a Cauchy matrix parametrized by two integer vectors, and the SVD of the Hilbert matrix $\mathcal{H}$ (or any other Cauchy matrix) of size say $n = 100$ and with $\kappa_2(\mathcal{H}) > 10^{150}$ can be computed to nearly full machine precision; see [13]. Following this line of reasoning, we recall the well-known connection between the Hankel and the Vandermonde matrices. For any Vandermonde matrix $\mathcal{V}$ and any diagonal matrix D, the matrix $\mathcal{H} = \mathcal{V}^T D\mathcal{V}$ is Hankel. Furthermore, any nonsingular Hankel matrix $\mathcal{H}$ can be written as $\mathcal{H} = \mathcal{V}^T D\mathcal{V}$ with a suitable Vandermonde matrix $\mathcal{V}$ and a diagonal one D:

$$\begin{bmatrix} h_1 & h_2 & h_3 & \cdot & h_n \\ h_2 & h_3 & \cdot & h_n & h_{n+1} \\ h_3 & \cdot & \cdot & h_{n+1} & \cdot \\ \cdot & h_n & h_{n+1} & \cdot & h_{2n-2} \\ h_n & h_{n+1} & \cdot & h_{2n-2} & h_{2n-1} \end{bmatrix} \tag{1.1}$$
$$= \begin{bmatrix} 1 & 1 & \cdot & 1 & 1 \\ x_1 & x_2 & \cdot & x_{n-1} & x_n \\ x_1^2 & x_2^2 & \cdot & x_{n-1}^2 & x_n^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_1^{n-1} & x_2^{n-1} & \cdot & x_{n-1}^{n-1} & x_n^{n-1} \end{bmatrix} \begin{bmatrix} d_1 & & & & \\ & d_2 & & & \\ & & \cdot & & \\ & & & d_{n-1} & \\ & & & & d_n \end{bmatrix} \begin{bmatrix} 1 & x_1 & x_1^2 & \cdot & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdot & x_2^{n-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_{n-1} & x_{n-1}^2 & \cdot & x_{n-1}^{n-1} \\ 1 & x_n & x_n^2 & \cdot & x_n^{n-1} \end{bmatrix}.$$

This decomposition naturally arises if the underlying finite-rank Hankel operator $\mathfrak{H} : \ell^2 \longrightarrow \ell^2$ is defined by its rational symbol that is represented in pole-residue form $\chi(z) = \sum_{j=1}^{n} \frac{d_j}{z - x_j}$. In that case, $\mathcal{V}$ is replaced by the infinite matrix $\mathcal{V}_\infty = (\mathcal{V}, D^n\mathcal{V}, D^{2n}\mathcal{V}, \ldots)$, and $\mathcal{V}^T D\mathcal{V}$ is the leading $n \times n$ submatrix of $\mathfrak{H} \equiv \mathcal{V}_\infty^T D\mathcal{V}_\infty$; see [30]. For theoretical implications of (1.1) and an extension to block Hankel matrices factored using confluent Vandermonde matrices, see [26]. In signal processing (Prony's method for frequency analysis), this corresponds to a signal given as a sum of exponentials, $h_j = \sum_{k=1}^{r} d_k x_k^j$, and $\mathcal{V}(x)$ is $r \times n$. For a given $\mathcal{H}$, revealing the modes that generated the signal (which is contaminated by noise) requires computing the Vandermonde decomposition (of a low-rank Hankel approximation to $\mathcal{H}$), and for that task, algorithms based on the nonsymmetric Lanczos method have been constructed; see [7]. We do not consider the difficult problem of computing the decomposition (1.1) of a given matrix $\mathcal{H}$. Instead, the starting point for the development of this paper is the assumption that $\mathcal{H}$ is given implicitly by the vectors $x, d \in \mathbb{C}^n$ as in (1.1).

From now on, we assume that $\mathcal{H}$ has been constructed from the entries of $\mathcal{V} = \mathcal{V}(x)$ and $D = \mathrm{diag}(d)$ with x and d given as raw data. Then, we take up the challenge of computing the SVD of $\mathcal{H} = \mathcal{V}^T D\mathcal{V}$ as a function of x and d. We allow $\mathcal{V}(x)$ to be an $r \times n$ matrix with

ETNA
Kent State University
http://etna.math.kent.edu

an $r \times r$ diagonal matrix D, and we do not assume any particular structure of x and d although in practice they might satisfy certain conditions. For instance: (i) an equivalent condition for the boundedness of $\mathfrak{H}$ is $\max_i |x_i| < 1$, or (ii) in system modeling, the data x_i, d_i may appear in complex conjugate pairs, which opens the possibility of computing only in real arithmetic.

Due to the potentially severe ill-conditioning of $\mathcal{V}$ and arbitrarily high condition number of D, an accurate SVD of the product $\mathcal{V}^T D\mathcal{V}$ is a nontrivial task. For example, the spectral condition number $\kappa_2(\mathcal{V})$ of an arbitrary real $n \times n$ Vandermonde matrix exceeds $2^{n-2}/\sqrt{n}$ (for instance, for $n = 200$, $\kappa_2(\mathcal{V}) > 2.8 \cdot 10^{58}$); see, e.g., [6, 48]. On the other hand, from the representation $\mathcal{H} = \mathcal{V}(x)^T \mathrm{diag}(d)\mathcal{V}(x) \equiv \mathcal{H}(x, d)$, we can compute the determinant $\det(\mathcal{H}) = (\prod_{i=1}^n d_i) \prod_{1 \le i < j \le n} (x_j - x_i)^2$ and $|\det(\mathcal{H})|$ with a small forward relative error for any input x, d. Hence, the necessary condition of being able to compute $|\det(\mathcal{H})| \equiv \sigma_1 \cdots \sigma_n$ accurately is satisfied, and trying to construct an accurate algorithm for computing $\sigma_1, \ldots, \sigma_n$ is not necessarily a futile effort.

Our main contribution in this work is the development of new algorithms and an introduction of a general technique to compute the SVD of the products $\mathcal{V}^T D\mathcal{V}$ and $\mathcal{C}^T \widetilde{D}\mathcal{C}$ (with Vandermonde $\mathcal{V}$, Cauchy $\mathcal{C}$, and arbitrary diagonal matrices D, $\widetilde{D}$) to high relative accuracy and with reliable and computable error bounds. In fact, we can compute the singular values in the whole range of positive floating-point numbers, even the tiniest ones at the level of the underflow threshold, to nearly machine precision in double-precision (16 digits) IEEE arithmetic and even for a condition number as high as 10^{616} (i.e., the smallest singular values nearly underflow, while the largest ones nearly overflow). It will not always be possible to guarantee high accuracy a priori, but it will be possible to give reliable error estimates based on the computed condition numbers at the cost of at most $O(n^3)$. This situation is similar to the SVD of Cauchy matrices [13], where the accuracy of the algorithm depends on the condition numbers of the unit triangular factors from the LDU decomposition with complete pivoting—those numbers are well behaved in practice, but proving it is an open problem.

The rest of the paper is organized as follows. In order to make this presentation self-contained and to facilitate an easier introduction and analysis of new algorithms, in Section 2 we first give a brief review of relevant issues related to an accurate computation of the SVD. In Section 3 we review the symmetric structure of the SVD of Hankel matrices. In Section 4 we first show how the discrete Fourier transform can be used to reduce the problem to computing the SVD of the Cauchy product $\mathcal{C}^T \widetilde{D}\mathcal{C}$, and then we introduce two algorithms for the SVD of $\mathcal{H}$. Section 5 offers a detailed error and perturbation analysis that provides reliable estimates of the accuracy of the computed decomposition. Numerical experiments in Section 4.3.1 and Section 5.4 show a perfect match with the theory. In Section 6 we discuss some generalizations and provide examples that show the importance of the conditions identified in the analysis in Section 5. Final remarks and further research challenges, motivated by Section 6, are given in Section 7.

2. Preliminaries: computing an accurate SVD. If $\widetilde{\sigma}_1 \ge \cdots \ge \widetilde{\sigma}_n \ge 0$ are the computed approximations of the singular values $\sigma_1 \ge \cdots \ge \sigma_n > 0$ of a full-rank matrix $A \in \mathbb{C}^{m \times n}$, $m \ge n$, we need information on the errors $\delta\sigma_i = \widetilde{\sigma}_i - \sigma_i$. In this section we discuss the key elements of bounding the relative error $\max_i |\delta\sigma_i|/\sigma_i$. For the sake of brevity, the corresponding analysis of the computed singular vectors is omitted.

2.1. Accurate SVDs. One of the key observations in the development of accurate SVD algorithms is that the condition number[1] $\kappa_2(A) = \|A\|_2 \|A^\dagger\|_2 = \sigma_{\max}(A)/\sigma_{\min}(A)$ is not always the true measure of sensitivity of the computed SVD, at least not for all algorithms

[1] Here $A^\dagger$, $\sigma_{\max}(A)$, $\sigma_{\min}(A)$ denote the pseudo-inverse, the largest, and the smallest singular value of A, respectively.

ETNA
Kent State University
http://etna.math.kent.edu

and not for all input data. The structure and the magnitude of the initial uncertainty in A and the backward error generated by the algorithm will determine what condition number will influence the computed result. For example, the bidiagonalization-based QR and the *divide and conquer* algorithms compute the SVD with a backward error ΔA such that $\|\Delta A\|_2/\|A\|_2$ is small. The relative error in the computed singular values is at best bounded by $\max_i |\delta\sigma_i|/\sigma_i \leq \kappa_2(A)\|\Delta A\|_2/\|A\|_2$. If the initial uncertainty $\Delta_0 A$ in A is also such that only $\|\Delta_0 A\|_2/\|A\|_2$ is small (i.e., the best we can say is that A is somewhere in a $\|\cdot\|_2$-ball around the true data matrix), then we can expect that $\kappa_2(A)$ will correctly estimate the accuracy of the computed SVD, and better results are not warranted by the data—one cannot and should not expect better accuracy.

In some applications, $A \in \mathbb{C}^{m\times n}$ can be written as $A = XD$, where X has full column rank and is well-conditioned in the traditional sense ($\kappa_2(X)$ is moderate), D is diagonal, possibly very ill-conditioned, and only X is subject to perturbations. For instance, X can represent a discretized kernel function of a Fredholm integral operator, and D carries the quadrature weights. Or, in some other situation, the column scaling D can simply be a consequence of changing the physical units used to express the data, and it certainly cannot and should not make the problem ill-conditioned. In that case, $A + \Delta A = (X + \Delta X)D = (\mathbb{I} + \Delta X X^\dagger)XD$, and the perturbation theory [25] implies that the induced perturbation in the singular values is bounded by the size of the perturbation $\Delta X X^\dagger$ of the identity $\mathbb{I}$:

$$\max_i |\delta\sigma_i|/\sigma_i \leq \kappa_2(X)\|\Delta X\|_2/\|X\|_2, \quad X = AD^{-1}, \quad \Delta X = \Delta A D^{-1}. \tag{2.1}$$

Let the diagonal entries of D be the Euclidean lengths of the corresponding columns of A, and thus X has unit columns. In that case, a result of van der Sluis [49] implies that $\kappa_2(X) \leq \sqrt{n}\min_{\Delta=\mathrm{diag}}\kappa_2(A\Delta)$, and thus $\kappa_2(X) \leq \sqrt{n}\kappa_2(A)$. Since it is possible that $\kappa_2(X) \ll \kappa_2(A)$, the bound (2.1) is never much worse, and it is potentially much better than the traditional bounds based on $\kappa_2(A)$. In this situation, a high condition number $\kappa_2(A)$ is considered as an artificial ill-conditioning, and we ought to compute the singular values with the error bound (2.1). This is possible by using the Jacobi SVD algorithm [17, 23, 24], which we briefly review in Section 2.2.

2.2. One sided Jacobi SVD: column-scaling invariant condition number. A key difference between various SVD algorithms lies in the structure of the backward errors they generate, and for that reason the Jacobi SVD is more accurate than the bidiagonalization-based QR or the *divide and conquer* algorithm [17]. This difference is in particular important if $A \in \mathbb{C}^{m\times n}$ can be written as $A = XD$, as we have discussed in Section 2.1 above. We now give the key points in the analysis of the Jacobi SVD algorithm and highlight the main principles of avoiding the effects of the artificial ill-conditioning caused by a diagonal scaling. We use a simple version of the Jacobi SVD algorithm outlined in Algorithm 1 below. (For full details, we refer to [23, 24].)

Consider the backward errors in Algorithm 1. To ease notation, let A be pre-permuted, $A := A\Pi$, $\Pi := \mathbb{I}$. Then, in the first step, we have

$$A + \delta A = \widetilde{Q}\begin{bmatrix}\widetilde{R}\\ 0\end{bmatrix}, \qquad \text{where} \quad \|\delta A(:,i)\|_2 \leq \epsilon_{QR}\|A(:,i)\|_2, \quad i = 1,\ldots,n,$$

and $\widetilde{Q}$ is numerically orthogonal. Hence, the backward error in each column of A is small with respect to that column. This is a consequence of the fact that in the QR factorization, the columns of A are individually multiplied by nearly unitary transformations. In the second step, we have $(\widetilde{R}^* + \delta\widetilde{R}^*)\widetilde{J} = \widetilde{V}\widetilde{\Sigma}$, where $\widetilde{J}$ and $\widetilde{V}$ are numerically unitary, $\widetilde{\Sigma}$ is diagonal with the computed singular values $\widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \cdots \geq \widetilde{\sigma}_n$ on its diagonal. Since the post-multiplications by Jacobi rotations that form $\widetilde{J}$ are independent transformations of the rows

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 1 $(\Sigma, U, V) = \mathrm{SVD}(A)$
(SVD of $A \in \mathbb{C}^{m\times n}$, $m \geq n$).

1: $A\Pi = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$;
{Givens or Householder QR factorization with the Businger-Golub column pivoting [9].}
2: $R^* J = V\Sigma$;
{One-sided Jacobi SVD algorithm [24, 50]; J is the product of Jacobi rotations.}
3: The SVD of A is : $A = Q \begin{bmatrix} J & 0 \\ 0 & \mathbb{I} \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} (\Pi V)^*$.

of $\widetilde{R}^*$, the backward error can be estimated as $\|\delta\widetilde{R}^*(i,:)\|_2 \leq \epsilon_J \|\widetilde{R}^*(i,:)\|_2$. Here both ϵ_{QR} and ϵ_J are $O(n)\widehat{\varepsilon}$. Altogether, the computed matrices $\widetilde{Q}$, $\widetilde{V}$, $\widetilde{J}$, $\widetilde{\Sigma}$ form a backward perturbed numerical SVD

$$A + \Delta A = \widetilde{Q} \begin{bmatrix} \widetilde{J}^{-*} & 0 \\ 0 & \mathbb{I} \end{bmatrix} \begin{bmatrix} \widetilde{\Sigma} \\ 0 \end{bmatrix} \widetilde{V}^*, \quad \Delta A = \delta A + \widetilde{Q} \begin{bmatrix} \delta\widetilde{R} \\ 0 \end{bmatrix},$$
$$\|\Delta A(:,i)\|_2 \leq (\epsilon_{QR} + \epsilon_J(1+\epsilon_{QR}))\|A(:,i)\|_2,$$

where in practice $\widetilde{J}$ replaces $\widetilde{J}^{-*}$ with a small error because $\widetilde{J}$ is numerically orthogonal. Note that $A = XD$ (with $D = \mathrm{diag}(\|A(:,i)\|_2)_{i=1}^n$) is perturbed to $XD + \Delta A = (X + \Delta A D^{-1})D$, where $\Delta X \equiv \Delta A D^{-1}$ is small in norm relative to X, and the bound (2.1) applies.

As a summary, the following basic facts will be taken to the next step in our development:

(i) The initial matrix is transformed as $A \rightsquigarrow A\Pi \rightsquigarrow Q^*(A\Pi) \rightsquigarrow \left[\begin{smallmatrix} J^* & 0 \\ 0 & \mathbb{I} \end{smallmatrix}\right] (Q^*(A\Pi))$, that is, all the time the (permuted) columns are being multiplied from the left independently. The only transformation from the right is a reordering of the columns of A, which is error-free. There has been no mixing of large and small columns. Such mixing of large and small columns carries the risk of losing information on the small singular values. (This is what can happen during the reduction to bidiagonal form.)

(ii) The pivoting using Π (following any suitable pivot strategy) is not essential for the point made in (i) above. However, pivoting has a strong preconditioning effect: R can be written as $R = ST$, where S is diagonal and T is usually much better conditioned than X. This greatly reduces the number of Jacobi rotations needed to reach numerical convergence. On the other hand, in Algorithm 2 in Section 2.4 below, the pivoting will be essential. The effect of Π on the structure of R is further discussed in Section 5.2.4.

REMARK 2.1. In practice, very often nearly the same accuracy is attained if, in line 2 of Algorithm 1, we compute the SVD of R^* using Matlab's $\mathtt{svd}(R^*)$ (instead of the one-sided Jacobi SVD). This can be partly explained using the results of Jesse Barlow [4].

2.3. Rank-revealing decompositions. Many classes of matrices are ill-conditioned without well-conditioned $A = XD$ representations, but it is possible to decompose them as $A = XDY^*$ (e.g., using LDU with complete pivoting), where X and Y are of full column rank, well-conditioned, and D is a possibly very ill-conditioned diagonal nonsingular matrix. Such a decomposition is called rank-revealing (RRD) [15].

Suppose that we know how to compute this decomposition in a forward stable way in the following sense: the computed matrices $\widetilde{X} = X + \delta X$, $\widetilde{Y} = Y + \delta Y$, and $\widetilde{D} = D + \delta D$ satisfy

$$\frac{\|\delta X\|_2}{\|X\|_2} \leq \epsilon, \quad \frac{\|\delta Y\|_2}{\|Y\|_2} \leq \epsilon, \quad \max_i \left|\frac{\delta D_{ii}}{D_{ii}}\right| \leq \epsilon. \tag{2.2}$$

ETNA
Kent State University
http://etna.math.kent.edu

Then, with a suitable ΔX, we can write

$$\widetilde{X}\widetilde{D}\widetilde{Y}^* = (\mathbb{I} + \Delta X X^\dagger) X D Y^* (\mathbb{I} + \delta Y Y^\dagger)^*, \quad \|\Delta X\|_2/\|X\|_2 \leq 2\epsilon + \epsilon^2. \tag{2.3}$$

For sufficiently small ΔX, ΔY, the perturbation estimate [25, Theorem 3.1] applied to (2.3) yields for each singular value σ_i and its computed approximation $\widetilde{\sigma}_i$,

$$\begin{aligned} &\sigma_i \left(1 - \kappa_2(X)\frac{\|\Delta X\|_2}{\|X\|_2}\right)\left(1 - \kappa_2(Y)\frac{\|\Delta Y\|_2}{\|Y\|_2}\right) \\ &\qquad \leq \widetilde{\sigma}_i \leq \sigma_i \left(1 + \kappa_2(X)\frac{\|\Delta X\|_2}{\|X\|_2}\right)\left(1 + \kappa_2(Y)\frac{\|\Delta Y\|_2}{\|Y\|_2}\right). \end{aligned} \tag{2.4}$$

The key observation here is that a forward stable decomposition avoids $\kappa_2(A)$ and replaces it essentially with $\max(\kappa_2(X), \kappa_2(Y))$ independent of the ill-conditioning in D. For special classes of matrices such a decomposition is provided by Gaussian eliminations with complete pivoting. The "DGESVD paper" [15] shows that under certain algebraic and combinatorial conditions, the pivoted LDU can be computed in a forward stable way so that (2.2) holds true with well-conditioned matrices X, Y. To illustrate this, let $A = \mathcal{C}$, where $\mathcal{C}_{ij} = 1/(\alpha_i + \beta_j)$ is a Cauchy matrix. In that case, an entry-wise forward stable pivoted LDU decomposition $P_1 \mathcal{C} P_2 = LDU$ can be computed using the parameters $(\alpha_i) \in \mathbb{C}^m$, $(\beta_j) \in \mathbb{C}^n$; see [13, 15]. The RRD $\mathcal{C} = (P_1^T L) D (U P_2^T) \equiv X D Y^*$ is computed as $\mathcal{C} \approx \widetilde{X}\widetilde{D}\widetilde{Y}^*$ with $|X_{ij} - \widetilde{X}_{ij}| \leq \epsilon |X_{ij}|$, $|D_{ii} - \widetilde{D}_{ii}| \leq \epsilon |D_{ii}|$, $|Y_{ij} - \widetilde{Y}_{ij}| \leq \epsilon |Y_{ij}|$, for all i, j, and ϵ of the order of $O(n)\widehat{\varepsilon}$. Hence, by virtue of (2.4), it remains to compute the SVD of the product $\widetilde{X}\widetilde{D}\widetilde{Y}^*$ with an accuracy governed by $\max(\kappa_2(\widetilde{X}), \kappa_2(\widetilde{Y}))$.

2.4. The PSVD. This approach of first decomposing A via the LDU and then computing its SVD from the product of the LDU factors [19, Section 3.6], [15, Section 2, 3] relies on the ability to compute the SVD of the RRD $A = XDY^*$ to high relative accuracy, determined by $\max(\kappa_2(X), \kappa_2(Y))$ and independent of $\kappa_2(D)$ or $\kappa_2(XDY^*)$. This task can be accomplished with the following algorithm [19]:

Algorithm 2 $(\Sigma, U, V) = \mathrm{PSVD}(X, D, Y)$
(SVD of XDY^*; $X \in \mathbb{C}^{m\times n}$, $Y \in \mathbb{C}^{p\times n}$).

1: $D_X = \mathrm{diag}(\|X(:,i)\|_2)_{i=1}^n$; $X_c = X D_X^{-1}$; $Y_1 = Y D^* D_X^*$;
{Internal scaling; $XDY^* = X_c Y_1^*$.}
2: $Y_1 \Pi = Q \left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right]$;
{Givens or Householder QR factorization with the Businger-Golub column pivoting [9].}
3: Compute the matrix $Z = X_c \Pi R^*$; {Explicit matrix multiplication.}
4: Compute the SVD of Z: $Z = U \Sigma V_z^*$; {SVD using the Jacobi SVD algorithm [23, 24].}
5: The SVD of XDY^* is : $XDY^* = U \left[\Sigma \quad 0\right] \left(Q \left[\begin{smallmatrix} V_z & 0 \\ 0 & \mathbb{I} \end{smallmatrix}\right]\right)^*$.

The pivoting in line 2 is essential—the resulting R can be factored as $R = ST$ with a diagonal matrix S and a well-conditioned one T, where we can choose T to have a unit diagonal or all rows of unit Euclidean norm. This allows an explicit computation of the matrix Z in line 3 and an accurate SVD computation of Z in line 4. It should be noted that $Z = (X_c \Pi) R^*$ must be computed with the standard $O(mn^2)$ matrix multiplication algorithm. Fast matrix multiplication methods [44] do not meet the criteria for high accuracy. Also, $Z \equiv (X_c \Pi) T^* S^*$ has the structure *well-conditioned* $\times$ *diagonal*, and the computed singular values obey an error bound analogous to (2.4). The theoretical condition number of

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 2 is $\kappa_2(S)\max(\kappa_2(X),\kappa_2(Y))$, where $\kappa_2(S)$ is small (see Section 5.2.4), and its influence is never experienced in numerical experiments. It can be removed at the price of more computations (see [15, Algorithm 3.2]) but with no practical advantage over Algorithm 2. In addition, it can be easily shown that the accuracy of Algorithm 2 is determined (up to a factor of $\sqrt{n}$) by the condition number $\kappa_2(S)\max(\min_{D=\mathrm{diag}}\kappa_2(XD),\min_{D=\mathrm{diag}}\kappa_2(YD))$.

3. Hankel matrices and symmetric (Autonne-Takagi) SVDs. The seemingly simple and aesthetically pleasing definition of Hankel matrices hides a rather rich structure with implications in matrix theory, rational approximation theory, signal processing, control theory, computational geometry, and algebraic coding theory. Various relations connect them to Toeplitz, Vandermonde, Bernstein, Cauchy, Pascal, Krylov, companion, and other structured matrices. Their inverses are Bezoutiants of certain polynomial pairs [28]. Hankel matrices of dimension $n\times n$ form a $2n-1$ dimensional linear subspace of $\mathbb{C}^{n\times n}$ which, together with Toeplitz matrices, has the unique property (among all subspaces of $\mathbb{C}^{n\times n}$ of dimension $2n-1$) that any $n\times n$ matrix can be expressed as a product of $4r+1$ Hankel matrices with $r=\lfloor n/2\rfloor+1$. For a nonsingular matrix, only $2r$ factors suffice; for a generic matrix (with exceptions on a proper subvariety), r factors are enough; see [53].

In addition, the symmetry $\mathcal{H}=\mathcal{H}^T$ implies that the SVD $\mathcal{H}=\mathbb{U}\Sigma\mathbb{V}^*$ has a special structure shared by all complex symmetric matrices: any $A=A^T\in\mathbb{C}^{n\times n}$ can be written in the form $A=\mathbb{W}\Sigma\mathbb{W}^T$ with a unitary matrix $\mathbb{W}$, and this is called the Takagi factorization [45] of A. It also appears in a work of Schur [42]. The next theorem provides the details of how an SVD of $A=A^T$ can be turned into the symmetric SVD, and it is included for the reader's convenience. It also serves as a tribute to the work of Léon Autonne, who introduced this decomposition in his 1915 paper [3, Chapitre VIII. Theoréme 92°, page 65].

THEOREM 3.1 (Autonne-Takagi factorization). *Let A be a complex symmetric matrix. Then its SVD can be written as $A=\mathbb{W}\Sigma\mathbb{W}^T$, where $\mathbb{W}$ is unitary and Σ is a diagonal matrix carrying the singular values of A.*

Proof. Here we follow [37, Section 3.1, Problem 33], which is based on the approach of Autonne [3, VIII]. From[2] $A=\mathbb{U}\Sigma\overline{\mathbb{V}}^T=\overline{\mathbb{V}}\Sigma\mathbb{U}^T=A^T$, it follows that $\mathbb{V}^T\mathbb{U}\Sigma=\Sigma(\mathbb{V}^T\mathbb{U})^T$, which implies that $\mathbb{V}^T\mathbb{U}$ is block diagonal with unitary symmetric blocks along the diagonal of sizes corresponding to the multiplicities of the singular values. More precisely, let $\Sigma=\oplus_{i=1}^p\sigma_i^\sharp\mathbb{I}_{m_i}$, where $\sigma_1^\sharp>\cdots>\sigma_p^\sharp\geq 0$ are all different singular values with multiplicities $m_1,\ldots,m_p$, respectively. Then $\mathbb{V}^T\mathbb{U}=\oplus_{i=1}^p K_i$, where $K_i^*K_i=\mathbb{I}$ and $K_i=K_i^T$ for $\sigma_i^\sharp\neq 0$. For each K_i, we can use its Schur form and construct a unitary square root S_i such that $S_i^2=K_i$. If we represent S_i using the Lagrange interpolating polynomial, we see that $S_i=S_i^T$ if $K_i=K_i^T$. Now, the unitary matrix $S=\oplus_{i=1}^p S_i$ satisfies $S^2=\mathbb{V}^T\mathbb{U}$ and $S\Sigma=\Sigma S=\Sigma S^T$; cf. [3, 91°, page 65; Lemme 39°, page 36]. Finally, we can write

$$\mathbb{U}\Sigma\mathbb{V}^*=\mathbb{U}\Sigma\overline{\mathbb{V}}^T=\overline{\mathbb{V}}\mathbb{V}^T\mathbb{U}\Sigma\overline{\mathbb{V}}^T=\overline{\mathbb{V}}SS\Sigma\overline{\mathbb{V}}^T=\overline{\mathbb{V}}S\Sigma S^T\overline{\mathbb{V}}^T=\mathbb{W}\Sigma\mathbb{W}^T,\quad \mathbb{W}\equiv\overline{\mathbb{V}}S.\qquad\square$$

It should be noted that an SVD of $A=A^T\in\mathbb{C}^{n\times n}$ is not automatically its Autonne-Takagi factorization, and that the recipe given in the proof of Theorem 3.1 is very simple in the case of simple singular values. For the sake of simplicity, we consider only the SVD. Note that $A\overline{\mathbb{W}}=\mathbb{W}\Sigma$ solves the con-eigenvalue problem for A. In fact, this con-(eigenvalue, eigenvector) property led Takagi [45] to discover the symmetric SVD as an algebraic tool for constructing rational approximations on the unit disk. It is also worth mentioning that the

[2] Here $\overline{\mathbb{V}}$ denotes entry-wise complex conjugation, so that $\mathbb{V}^*=\overline{\mathbb{V}}^T$.

ETNA
Kent State University
http://etna.math.kent.edu

singular values of A have a variational characterization in terms of the bilinear form $v^T Av$, $v \in \mathbb{C}^n$ [11] (in addition to the min-max characterization as the roots of the eigenvalues of A^*A).

The Hankel structure allows computing the (symmetric) SVD of $\mathcal{H}(h)$ with an $O(n^2 \log n)$ complexity; see [52]. However, since our primary goal is guaranteed high numerical accuracy in all singular values and singular vectors, we are willing to trade efficiency for accuracy and to accept the standard $O(n^3)$ complexity of an algorithm if it is capable of delivering even the tiniest singular values to nearly full machine precision.

4. Algorithms for an accurate SVD of $\mathcal{H} \equiv \mathcal{V}^T D\mathcal{V}$. We now describe an algorithm to compute the SVD of the product $\mathcal{H} = \mathcal{V}^T D\mathcal{V} \in \mathbb{C}^{n\times n}$, where $\mathcal{V}$ is Vandermonde and D is diagonal given by the vectors $x, d \in \mathbb{C}^n$, respectively. The key step is a reparametrization of the product $\mathcal{V}^T D\mathcal{V}$ by replacing the Vandermonde matrix $\mathcal{V}$ by a Cauchy matrix, and it is motivated by the availability of an RRD for Cauchy matrices.

4.1. A prelude: the DFT of the Vandermonde product form. It is well known that any Vandermonde matrix $\mathcal{V}_n(x) = \mathcal{V} = (x_i^{j-1})_{i,j=1}^n$ can be written as $\mathcal{V} = D_1\mathcal{C}D_2\mathbb{F}^*$, where $\mathbb{F}$ is the unitary $n \times n$ FFT matrix ($\mathbb{F}_{ij} = \omega^{(i-1)(j-1)}/\sqrt{n}$, $\omega = \mathbf{e}^{2\pi \mathrm{i}/n}$), D_1 and $\mathcal{D}_2$ are diagonal, and $\mathcal{C}$ is a Cauchy matrix. More precisely,

$$(\mathcal{V}\mathbb{F})_{ij} = \left[\frac{1-x_i^n}{\sqrt{n}}\right]\left[\frac{1}{\omega^{1-j}-x_i}\right]\left[\frac{1}{\omega^{j-1}}\right] \equiv (D_1)_{ii}\,\mathcal{C}_{ij}\,(\mathcal{D}_2)_{jj}, \quad 1 \le i,j \le n. \tag{4.1}$$

This is turned into an accurate SVD of $\mathcal{V}_n(x)$, but, with quite a few fine details, tuned to perfection in [13, 16]. In particular, a possible singularity at x_i equal to a floating-point value of an nth root of unity is removable.

In our setting, this gives $\mathbb{F}^T\mathcal{H}\mathbb{F} = \mathcal{D}_2\mathcal{C}^T D_1 D D_1 \mathcal{C}\mathcal{D}_2$. Since $\mathbb{F}$ and $\mathcal{D}_2$ are unitary, it remains to compute the SVD of the complex symmetric matrix $M = \mathcal{C}^T D_3 \mathcal{C}$, where $D_3 = D_1^2 D$. Note that M is given implicitly and its factors $\mathcal{C}$ (given implicitly in (4.1)) and D_3 are given with full accuracy. (For details, see [13].)

4.2. The SVD of the Cauchy product form and the GRRD. We now focus on the SVD of $M = \mathcal{C}^T D_3 \mathcal{C}$. Let $\sqrt{D_3}$ be a square root such that $\sqrt{D_3}\sqrt{D_3} = D_3$, and let $\sqrt{D_3}\mathcal{C} = X D_4 Y^T$ be an RRD of $\sqrt{D_3}\mathcal{C}$, computed, e.g., from the LU decomposition of $\Pi_1\sqrt{D_3}\mathcal{C}\Pi_2 = LD_4U$. Let $X = \Pi_1^T L$ and $Y = \Pi_2 U^T$. Then

$$M = \mathcal{C}^T(\sqrt{D_3})^2\mathcal{C} = Y D_4 X^T X D_4 Y^T = \Pi_2 N \Pi_2^T, \qquad N = U^T D_4 L^T L D_4 U, \tag{4.2}$$

and the problem reduces to calculating the SVD of N. Note that in floating-point arithmetic, all three factors L, U, and D_4 are computed to high relative accuracy (they are entry-wise forward stable functions of x and d), and that L and U are unit triangular with off-diagonal entries less than one in modulus and expected to be well-conditioned, thus both $\kappa_2(L)$ and $\kappa_2(U)$ will be moderate [13]. We postpone the discussion on condition numbers to Section 5.1. Let us for a moment assume that $Z \equiv L^T L$ is well-conditioned to allow for an accurate LDU decomposition and that the diagonal entries of D_4 are monotonically decreasing in absolute values. Then, the product $N = U^T(D_4 Z D_4)U$ is a generalization of the RRD. Instead of the decomposition *well-conditioned* $\times$ *diagonal* $\times$ *well-conditioned*, we have *well-conditioned* $\times$ *matrix with accurate RRD* $\times$ *well-conditioned,* which is reduced to RRD by a pivoted LDU (or pivoted QR factorization) of the middle matrix $D_4 Z D_4$ in the product. This more general RRD (GRRD) has been introduced and analyzed in detail in [20].

ETNA
Kent State University
http://etna.math.kent.edu

4.3. New algorithms. So far, we have $D_2^* \mathbb{F}^T \mathcal{H} \mathbb{F} D_2^* = \Pi_2 N \Pi_2^T$, and it remains to compute the SVD of $N = U^T (D_4 L^T L D_4) U$. Based on the previous considerations, we can expect that the LDU decomposition with complete pivoting of $A = D_4 Z D_4$ will be an accurate RRD, $A = X_A D_A Y_A^*$, and N in (4.2) can be written as $N = (U^T X_A) D_A (Y_A^* U)$. The SVD in this form can be computed by Algorithm 2. Hence, putting all this together, we can formulate Algorithm 3.

Algorithm 3 $(\Sigma, \mathbb{U}, \mathbb{V}) = \mathrm{SVD\_VTDV}_0(x, d)$
(SVD of $\mathcal{H}(x,d) \equiv \mathcal{V}(x)^T \mathrm{diag}(d) \mathcal{V}(x)$, $x, d \in \mathbb{C}^n$).

1: $\omega = \mathbf{e}^{2\pi \mathbf{i}/n}$; $\vec{\omega} = (\omega^{1-i})_{i=1}^n$; $d_1 = (1 - x.^n)/\sqrt{n}$; $d_2 = (\omega^{j-1})_{j=1}^n$;
2: $[L, d_4, U, \pi_1, \pi_2] = \mathrm{CauchyLDU}(-x, \vec{\omega}, d_1 .* \sqrt{d}, ones(n,1))$;
{LDU of (4.1), following [13, 16].}
3: $A = (L\mathrm{diag}(d_4))^T L(\mathrm{diag}(d_4))$; {Diagonal scaling and cross-product.}
4: $[X_A, d_A, Y_A, \pi_3, \pi_4] = \mathrm{XDY}^*(A)$; {Here $A = X_A \mathrm{diag}(d_A) Y_A^*$.}
5: $[Q, R, \pi_5] = \mathrm{qr}(U^T \overline{Y_A} \mathrm{diag}(d_A))$;
{QR factorization with the Businger-Golub column pivoting.}
6: $S = U^T X_A(:, \pi_5) R^T$; {Lines 5–7 are from Algorithm 2.}
7: Compute the SVD of S: $S = U_s \Sigma W_s^*$; {SVD using the Jacobi SVD algorithm [23, 24].}
8: Assemble the singular vectors: $\mathbb{U} = \overline{\mathbb{F}} \mathrm{diag}(d_2) \Pi_2 U_s$; $\mathbb{V} = \mathbb{F} \mathrm{diag}(\overline{d_2}) \Pi_2 \overline{Q} W_s$.

It is instructive to state another algorithm as an illustration of the power of this technique and as a case study to point out some fine details as well as for some practical reasons; see Section 4.3.1. Algorithm 4 is based on the capability of the QR factorization with complete pivoting [41] to almost reach the stability of the LDU with complete pivoting [10, 21, 36] when applied to graded matrices, i.e., matrices of the form of $A = D_4 Z D_4$. The key is a pre-sorting of the rows of the input matrix in the vector ℓ_∞-norm denoted throughout the paper by $\|\cdot\|_\infty$.

Algorithm 4 $(\Sigma, \mathbb{U}, \mathbb{V}) = \mathrm{SVD\_VTDV}_1(x, d)$
(SVD of $\mathcal{H}(x,d) \equiv \mathcal{V}(x)^T \mathrm{diag}(d) \mathcal{V}(x)$, $x, d \in \mathbb{C}^n$).

1: $\omega = \mathbf{e}^{2\pi \mathbf{i}/n}$; $\vec{\omega} = (\omega^{1-i})_{i=1}^n$; $d_1 = (1 - x.^n)/\sqrt{n}$; $d_2 = (\omega^{j-1})_{j=1}^n$;
2: $[L, d_4, U, \pi_1, \pi_2] = \mathrm{CauchyLDU}(-x, \vec{\omega}, d_1 .* \sqrt{d}, ones(n,1))$;
{LDU of (4.1), following [13, 16].}
3: $[Q_1, R_1, \pi_3] = qr(U^T \mathrm{diag}(d_4))$;
{QR factorization with the Businger-Golub column pivoting.}
4: $G_0 = L(:, \pi_3) R_1^T$; $G = G_0^T G_0$; {Triangular matrix multiply and cross-product.}
5: π_4 is a sorting permutation such that $\|G(\pi_4(1), :)\|_\infty \geq \|G(\pi_4(2), :)\|_\infty \geq \cdots \geq \|G(\pi_4(n), :)\|_\infty$;
6: $[Q_2, R_2, \pi_5] = \mathrm{qr}(G(\pi_4, :))$;
{QR factorization with the Businger-Golub column pivoting.}
7: Compute the SVD of R_2^*: $R_2^* = U_r \Sigma W_r^*$; {SVD using the Jacobi SVD algorithm [23, 24].}
8: Singular vectors:
$\mathbb{U} = \mathbb{F}^{-1} \mathrm{diag}(d_2) Q_1(\pi_2^{-1}, :) Q_2(\pi_4^{-1}, :) W_r$; $\mathbb{V} = \mathbb{F} \mathrm{diag}(\overline{d_2}) \overline{Q}_1(\pi_2^{-1}, :) U_r(\pi_5^{-1}, :)$.

REMARK 4.1. Both Algorithm 3 and Algorithm 4 can be easily adapted to work with rectangular fat $\mathcal{V}$ with essentially the same accuracy properties. For the sake of brevity, we discuss only the square case and refer to [19, 20] for the details needed for the rectangular case.

ETNA
Kent State University
http://etna.math.kent.edu

4.3.1. Numerical examples. Let us now take the above developed algorithms for a test drive.[3] Clearly, there is a technical difficulty in obtaining reference values if the condition number of the matrix is as big as, say, 10^{500}. In that case, we can resort to special toolboxes with variable precision arithmetic and take as many digits of precision as necessary to compensate the high condition number. Another, indirect way to check the accuracy of the results is by comparing the outputs of two different algorithms that compute highly ill-conditioned decomposition—if they agree to almost full machine precision, a kind of Ockham's razor would deem them both accurate. A numerical analyst's way is to compute all relevant condition numbers and plug them into the state of the art error analysis and perturbation theory. For the sake of better understanding and demonstration of the robustness of the new algorithms, we combine all three methods.

EXAMPLE 4.2. With $n = 160$, we generate random complex vectors $x, d \in \mathbb{C}^n$ with normally distributed real and imaginary parts. The implicitly defined Hankel matrices $\mathcal{H} = \mathcal{V}(x)^T \mathrm{diag}(d)\mathcal{V}(x)$ in these experiments have, with varying x and d, spectral condition numbers of the orders of magnitude 10^{260} to 10^{280}. Hence, 300-digit arithmetic would be necessary to guarantee standard double-precision (16 digits) accuracy in a conventional SVD computation. Let us show the results of one concrete example with a condition number estimated as $\kappa_2(\mathcal{H}) \approx 1.4095\text{e+}260$.

For starters, let us compare in Figure 4.1 the singular values computed by Algorithm 3 ($\sigma_1^{(0)} \geq \cdots \geq \sigma_n^{(0)}$ are shown as blue circles) with those obtained by Algorithm 4 ($\sigma_1^{(1)} \geq \cdots \geq \sigma_n^{(1)}$ are shown as red $\times$'s, each scoring a bull's-eye on the blue circle). We observe perfect matches from the largest all the way down to the tiniest singular values. The computed relative differences $\epsilon_j^{(0,1)} = |\sigma_j^{(0)} - \sigma_j^{(1)}|/\sqrt{\sigma_j^{(0)}\sigma_j^{(1)}}$ are all below 1.0403e-013, which is almost $n \cdot \mathrm{eps} \approx 3.5527\text{e-}014$. This is indicative but not a proof that the two algorithms have delivered the singular values correct to almost full machine precision.[4] To verify the high accuracy of both algorithms, we computed the SVD of $\mathcal{H}$ in 300-digit arithmetic using the *Advanpix Multiprecision Computing Toolbox* for Matlab [1]. The computed singular values $\sigma_1 \geq \cdots \geq \sigma_n$ are used as reference values to calculate the errors $\epsilon_j^{(k)} = |\sigma_j - \sigma_j^{(k)}|/\sigma_j$, $k = 0, 1$. We obtained $\max_j \epsilon_j^{(0)} \leq 4.4405\text{e-}013$ and $\max_j \epsilon_j^{(1)} \leq 4.6522\text{e-}013$. It is remarkable that both Algorithm 3 and Algorithm 4 succeeded to compute all singular values ranging from $\sigma_1 \approx 5.9126\text{e+}139$ to $\sigma_n \approx 4.1949\text{e-}121$ to 13 digits of accuracy in ordinary double-precision computation with $\widehat{\varepsilon} \equiv \mathrm{eps} \approx 2.2204\text{e-}016$. The run times of the two algorithms[5] to find the full SVD (including singular vectors) were 1.80 (Algorithm 3) and 5.49 seconds (Algorithm 4), while computing with the toolbox in 300-digit arithmetic took $2.43 \cdot 10^4$ seconds.[6]

REMARK 4.3. We should note that, in the case when $\mathcal{H}$ is computed explicitly in floating-point arithmetic as $\widetilde{\mathcal{H}}$, it is possible that all singular values are computed with no accurate digit (i.e., all are wrong as compared to the true eigenvalues of $\mathcal{H}$) even if they are computed exactly from $\widetilde{\mathcal{H}}$. In Figure 4.1, we also display the computed singular values when Matlab's function

[3] *Beware of bugs in the above code; I have only proved it correct, not tried it.* – Donald Knuth

[4] Two different computational algorithms are obtaining nearly the same result of a highly ill-conditioned computation. See Example 6.2 for two algorithms getting almost identical and wrong result.

[5] On a Intel (R) Core (TM) 2 Duo CPU T6670 @ 2.20Ghz 2.20 Ghz based Laptop with 8GB RAM running Matlab under MS Windows 7. The algorithms are written as simple m-files without any attempt to optimize the run time.

[6] We have learned recently that new releases of the Advanpix toolbox provide improved SVD methods that reduce the run time considerably, with a factor as large as thousand. However, no matter how optimized, extended (300-digit) arithmetic is necessarily more expensive than the standard double-precision computation capable of delivering working-precision accuracy of the SVD.

ETNA
Kent State University
http://etna.math.kent.edu

`svd()` and a Jacobi SVD algorithm are applied to $\widetilde{\mathcal{H}}$, where $\widetilde{\mathcal{H}}$ is obtained by rounding to double precision the matrix $\mathcal{H}$ computed explicitly in 300-digit arithmetic.

Let us check the singular vectors $U^{(0)} = (u_1^{(0)}, \ldots, u_n^{(0)})$ and $V^{(0)} = (v_1^{(0)}, \ldots, v_n^{(0)})$ found by Algorithm 3. Figure 4.2 displays the differences $\|u_i - u_i^{(0)}\|_2$, $\|v_i - v_i^{(0)}\|_2$, where the reference values u_i, v_i are computed in 300-digits arithmetics. Note that the measured errors perfectly match the dependence on the relative gaps in the spectrum as predicted by the perturbation theory [38]. Almost identical results for the left and the right singular vectors are the consequence of the structure described in Theorem 3.1. Although the matrices $U^{(0)}$ and $V^{(0)}$ are computed independently using different formulas, an a posteriori check shows that $(U^{(0)})^T V^{(0)}$ is a diagonal unitary matrix with an error less than 6.6569e-014; cf. the proof of Theorem 3.1.

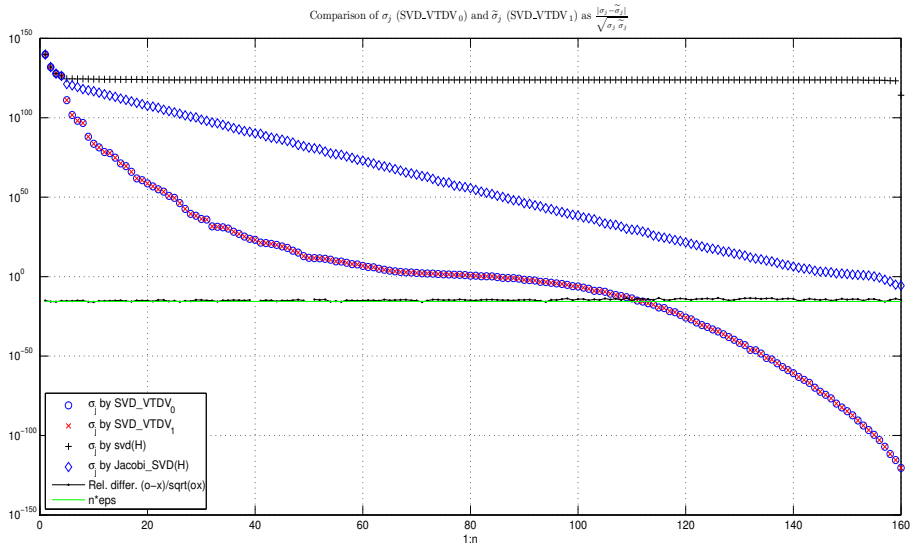

FIG. 4.1. *The singular values of $\mathcal{H} = \mathcal{V}^T D \mathcal{V} \in \mathbb{C}^{160\times 160}$ computed by Algorithm 3 (σ_j) and Algorithm 4 ($\widetilde{\sigma}_j$). The relative differences $|\sigma_j - \widetilde{\sigma}_j|/\sqrt{\sigma_j \widetilde{\sigma}_j}$ are all at the level of $n\cdot$eps, where $\widehat{\varepsilon} \equiv$ eps $\approx$ 2.2204e-016. Here $\kappa_2(\mathcal{H}) \approx$ 1.4095e+260.*

For the accuracy of the individual entries of the singular vectors, there is one a priori limiting factor—-the final FFT in line 8. Namely, if the FFT matrix $\mathbb{F}$ is applied to a vector x, then the result computed in floating-point arithmetic with the roundoff $\widehat{\varepsilon}$ reads [7]

$$\begin{aligned} &\mathfrak{fl}(\mathbb{F}x) = \mathbb{F}x + \epsilon, \\ &|\epsilon| \le \xi |\mathbb{F}||x| = \xi \frac{\|x\|_1}{\sqrt{n}} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \le \xi \|x\|_2 \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \text{where } \xi \le \frac{n\widehat{\varepsilon}}{1 - n\widehat{\varepsilon}}. \end{aligned} \tag{4.3}$$

Since the final step in the assembling of the singular vectors is the FFT of numerically unitary matrices, the bound (4.3) implies that we can expect a flat $O(n\widehat{\varepsilon})$-error over all entries of $U^{(0)}$ and $V^{(0)}$. Hence, the singular vector entries below $O(n\widehat{\varepsilon})$ will not be computed accurately. This is illustrated in Figure 4.3. The problem is not resolved by applying $\overline{\mathbb{F}}\mathrm{diag}(d_2)\Pi_2$ and

[7] The absolute values and inequalities involving vectors and matrices are understood entry-wise.

ETNA
Kent State University
http://etna.math.kent.edu

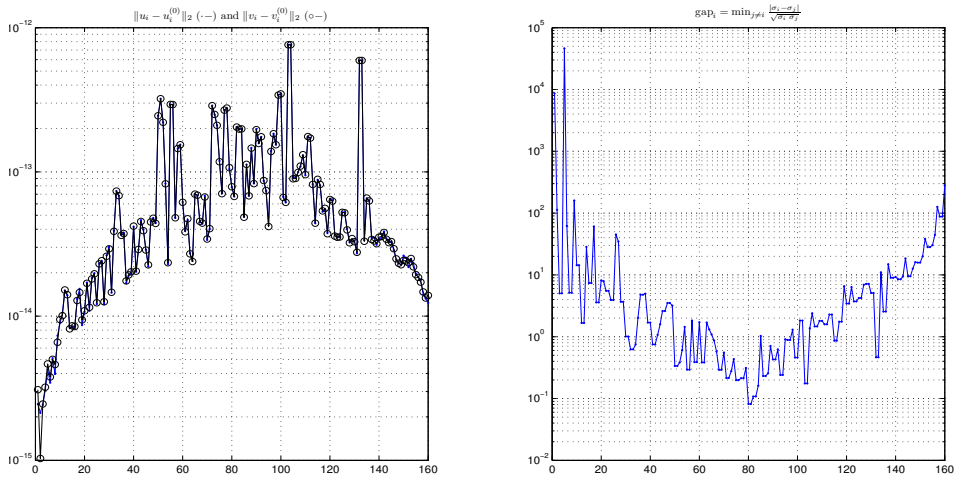

FIG. 4.2. *The accuracy of the singular vectors $u_i^{(0)}$, $v_i^{(0)}$ of $\mathcal{H} = \mathcal{V}^T D \mathcal{V} \in \mathbb{C}^{160\times 160}$ computed by Algorithm 3. The reference values are computed in 300-digits arithmetic. The plot on the right-hand side shows the relative gaps in the singular spectrum, $\mathrm{gap}_i = \min_{j\neq i} \frac{|\sigma_i-\sigma_j|}{\sqrt{\sigma_i\sigma_j}}$. Note that the measured errors comply with the perturbation theory. It holds that $\|u_i^{(0)} - u_i\|_2 \cdot \mathrm{gap}_i/(n\widehat{\varepsilon}) \in (1.54\cdot 10^{-1}, 5.57\cdot 10^{3})$ and $\|v_i^{(0)} - v_i\|_2 \cdot \mathrm{gap}_i/(n\widehat{\varepsilon}) \in (1.55\cdot 10^{-1}, 6.08\cdot 10^{3})$. The $O(10^3)$ right boundary of this interval hides the condition number; see Remark 5.1.*

$\mathbb{F}\mathrm{diag}(\overline{d_2})\Pi_2$ immediately to the RRD factors of $\mathcal{C}$ instead of applying them in the final assembly of the singular vectors.

5. Error and perturbation analysis. We now give some details on the error and perturbation analysis. For brevity, we consider only Algorithm 3. Clearly, proving backward stability in terms of the input data x and d is out of question. Similarly, constructing a backward error $\delta\mathcal{H}$ with a Hankel structure is not feasible, and even if it was, it would be of little use because the matrix $\mathcal{H}$ is extremely ill-conditioned and no useful error bound would follow. So, the complete finite-precision computation will not be represented by the classical commutative diagram of a backward stable computational process.

Instead, we go ahead with a mixed forward/backward error analysis relying on the fact that particular steps of the algorithm have preconditioning effects. To illustrate this, in Section 5.1 we make a guided tour through the algorithm and identify the key condition numbers for each particular step. Then, in Section 5.2, we analyze the effects of finite-precision arithmetic. One of the key feature that will become apparent is that all extreme ill-conditioning is revealed in the diagonal scaling matrices $\mathrm{diag}(d_4)$, $\mathrm{diag}(d_A)$ and that the condition numbers of those diagonal matrices never enter the forward error bounds, thus this will be experienced by the algorithm as artificial ill-conditioning. This section can be considered as a detailed proof of a theorem that gives forward error bounds for the computed singular values.

5.1. On condition numbers. Let us discuss the condition numbers of the intermediate matrices computed in Algorithm 3. Since we aim at a small forward error, it is of vital importance that those numbers are moderate. For some of them we can derive theoretical bounds; sometimes overwhelming numerical evidence and heuristics imply that they are expected to be moderate. In any case, we can estimate them and have computable error bounds. If we want high accuracy to some pre-specified level, those numbers will provide information on how many extra digits are needed in a particular step to guarantee the required accuracy.

ETNA
Kent State University
http://etna.math.kent.edu

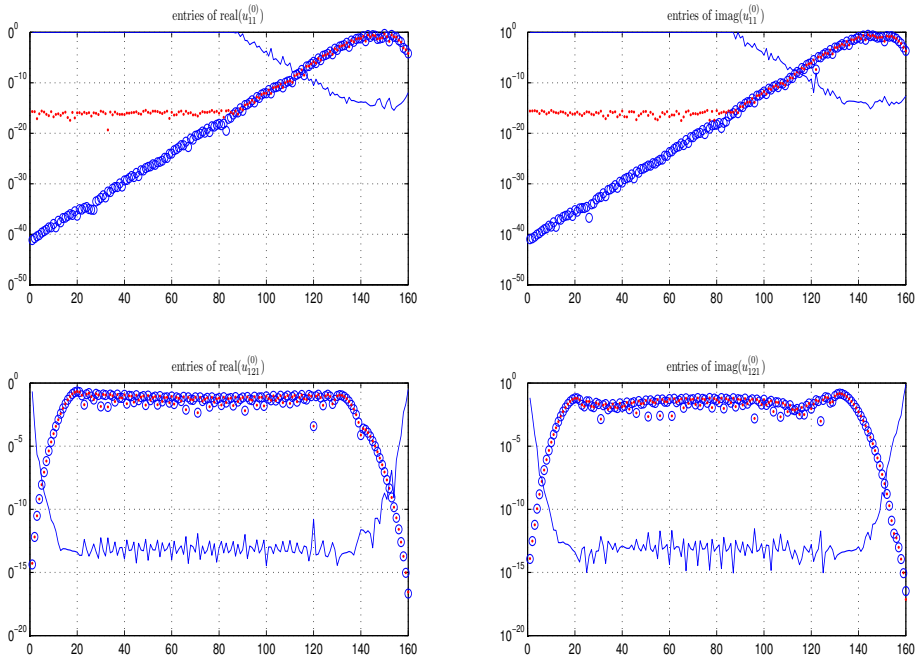

FIG. 4.3. *The accuracy of the singular vectors $u_{11}^{(0)}$, $u_{121}^{(0)}$ (entries plotted using "·") computed by Algorithm 3. The reference values ("○") are computed in* 300*-digits arithmetic. Accurately computed singular vectors' components are recognized as ⊙. The (blue) curve represents the relative errors in the singular vectors' components. The first two plots clearly show the effect of averaging of the FFT (4.3) and smearing the roundoff over the small components.*

5.1.1. Lines 2 and 3. The FFT of $\mathcal{V}$ is performed implicitly by defining the resulting Cauchy matrix $D_1\mathcal{C}D_2$. The first and the key step of removing the ill-conditioning is the pivoted LDU of $\sqrt{D_3}\mathcal{C}$. The resulting unit triangular matrices L and U can be considered well-conditioned, which is the case in all numerical experiments; see also [13, 15, 16]. For instance, in Example 4.2 we have $\kappa_2(L) \approx$ 6.6122e+001, $\kappa_2(U) \approx$ 1.7342e+002, and $\kappa_2(D_4) \approx$ 4.5508e+128. From the perturbation theory for singular values it follows that $D_4 = \mathrm{diag}(d_4) = L^{-1}\Pi_1\sqrt{D_3}\mathcal{C}\Pi_2 U^{-1}$ must be ill-conditioned. If $\sigma_i(\sqrt{D_3}\mathcal{C})$, $i = 1, \ldots, n$, are decreasingly ordered singular values of $\sqrt{D_3}\mathcal{C}$, then, with certain permutation ς, the results in [25] yield

$$\frac{\sigma_i(\sqrt{D_3}\mathcal{C})}{\|L\|_2\|U\|_2} \le |d_4|_{\varsigma(i)} \le \sigma_i(\sqrt{D_3}\mathcal{C})\|L^{-1}\|_2\|U^{-1}\|_2, \quad i = 1, \ldots, n.$$

It should be noted that moderate values of $\kappa_2(L)$ and $\kappa_2(U)$ are not just simple consequences of the fact that both are unit triangular matrices and that the remaining entries in their nontrivial triangles are bounded by one in modulus. The reasons are much deeper and beyond our full understanding at this moment; see [47, 51] for an in-depth analysis and discussion of the phenomena related to condition numbers of triangular matrices. Intuition indicates that Cauchy matrices are well separated from the apparently thin set of matrices with badly conditioned normalized triangular (pivoted) LDU factors. However, a thorough mathematical explanation of this observed phenomenon remains a challenging open problem. The best we can do in practical computation is to compute those condition numbers a posteriori and confirm that both matrices are well-conditioned. This can be done with an (acceptable) $O(n^2)$ complexity, and it should be included in a software implementation.

ETNA
Kent State University
http://etna.math.kent.edu

The same observation on the condition numbers holds true for the matrices X_A and Y_A, which are just permuted triangular matrices from the LDU decomposition with complete pivoting of $A = D_4 L^T L D_4$: $A(\pi_3, \pi_4) = L_A D_A U_A$, $X_A = L_A(\pi_3^{-1}, :)$, $Y_A = U_A(:, \pi_4^{-1})$. For the data of Example 4.2, we have $\kappa_2(X_A) \approxeq 4.1430\text{e+}000$, $\kappa_2(Y_A) \approxeq 4.1365\text{e+}000$, and $\kappa_2(D_A) \approxeq 2.7626\text{e+}257$.

5.1.2. Line 4. The computation of the LDU decomposition of $A = D_4 L^T L D_4$ in line 4 requires a more detailed discussion. The diagonal matrix D_4 can be very ill-conditioned with diagonal entries spanning over many orders of magnitude, expected to decrease in modulus from very large to very small (see the lower plot in Figure 5.1 for an example), although the strict monotonicity cannot be guaranteed. Typically, if the monotonicity is violated, the two positions where it happens are nearly of the same order of magnitude. This means that $A = D_4 Z D_4$ has a graded structure, which is in favor of an accurate LDU decomposition provided that the matrix $Z = L^T L$ is appropriately well-conditioned for the LDU computation; for a detailed discussion, see [15, Section 4]. It is immediate that $\kappa_2(Z) \leq \kappa_2(L)^2$ is expected to be small, but that is not enough—we need all leading submatrices of Z to be well-conditioned.

In our example, the upper plot on Figure 5.1 displays the values of $\kappa_2(Z(1:i, 1:i))$, $i = 1, \ldots, n$. They are all bounded by $\kappa_2(L)^2 \approxeq 4.3721\text{e+}003$. The permutations π_3, π_4 are equal and close to identity with only a few swaps of neighboring indices and do not change the structure illustrated in Figure 5.1. For the sake of experiment, we permuted the rows and the columns of Z with two random permutations and then recomputed the condition numbers of the leading submatrices. They were almost all above 10^{15} with some values even above 10^{20}. In another experiment, inspired by [47], we randomly changed the signs of the real and imaginary parts of L, which resulted in an increase of some values of $\kappa_2(Z(1:i, 1:i))$ above 10^8.

Apparently, the key for this well-behaved condition numbers lies in the fact that L has been computed by completely pivoted Gaussian eliminations of a Cauchy matrix; recall the discussion in Section 5.1.1. Unfortunately, making this theoretically more precise and proving the type of boundedness observed in numerical experiments does not seem to be simple. However, some useful facts about the condition numbers of certain submatrices of Z can be proved. Take some index i and partition the matrix Z as

$$Z = \begin{array}{c} \\ i \\ n-i \end{array} \begin{array}{c} \begin{array}{cc} i & n-i \end{array} \\ \begin{bmatrix} Z_{[11]} & Z_{[12]} \\ Z_{[21]} & Z_{[22]} \end{bmatrix} \end{array} = \begin{bmatrix} L_{[11]}^T & L_{[21]}^T \\ 0 & L_{[22]}^T \end{bmatrix} \begin{array}{c} \begin{array}{cc} i & n-i \end{array} \\ \begin{bmatrix} L_{[11]} & 0 \\ L_{[21]} & L_{[22]} \end{bmatrix} \end{array}.$$

Since $Z_{[22]} = L_{[22]}^T L_{[22]}$, it follows that $\kappa_2(Z_{[22]}) \leq \kappa_2(L_{[22]})^2 \leq \kappa_2(L)^2$. Hence, all trailing submatrices of Z are expected to be well-conditioned. Further, by the interlacing theorem for singular values, we know that the condition numbers of $Z_{[1]} = \begin{bmatrix} Z_{[11]} \\ Z_{[21]} \end{bmatrix}$ and $Z_{[2]} = \begin{bmatrix} Z_{[12]} \\ Z_{[22]} \end{bmatrix}$ are bounded above by $\kappa_2(Z) (\leq \kappa_2(L)^2)$. We are in fact interested in the condition number of the leading submatrix $Z_{[11]}$, but, unfortunately, the expression

$$Z_{[11]} = L_{[11]}^T L_{[11]} + L_{[21]}^T L_{[21]} = L_{[11]}^T (\mathbb{I} + L_{[11]}^{-T} L_{[21]}^T L_{[21]} L_{[11]}^{-1}) L_{[11]}$$

does not imply a similar conclusion except in the case of a real matrix L (which we do not have here). The problem is that, in the case of complex data, $L_{[11]}^T L_{[11]} + L_{[21]}^T L_{[21]}$ behaves quite differently from $L_{[11]}^* L_{[11]} + L_{[21]}^* L_{[21]}$. For the first summand, we have $\kappa_2(L_{[11]}^T L_{[11]}) \leq \kappa_2(L)^2$, but bounding the influence of the second term is not easy. An

ETNA
Kent State University
http://etna.math.kent.edu

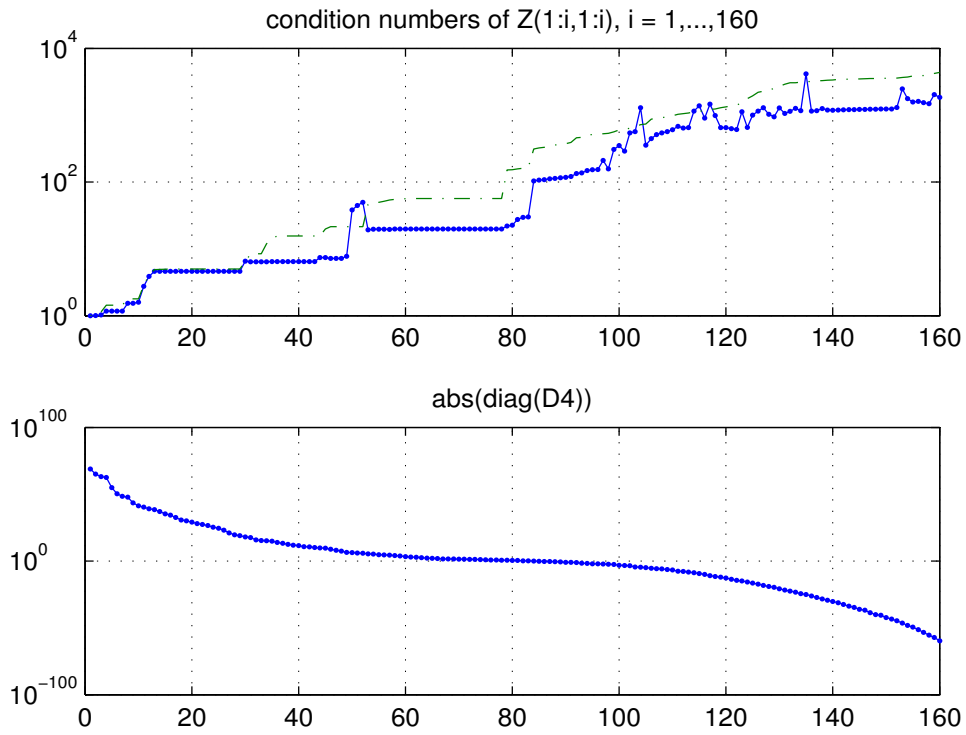

FIG. 5.1. *The condition numbers of the leading submatrices of the matrix $Z = L^T L$ and the diagonal scaling matrix D_4 for the Hankel matrix $\mathcal{H}$ from Example 4.2. The dashed line on the upper plot shows the condition numbers of the leading submatrices of $L^* L$; those numbers are, of course, provably non-decreasing.*

extremely ill-behaved (though contrived) example is $\begin{bmatrix} \mathbb{I} & \mathrm{i}\mathbb{I} \\ 0 & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbb{I} & 0 \\ \mathrm{i}\mathbb{I} & \mathbb{I} \end{bmatrix} = \begin{bmatrix} 0 & \mathrm{i}\mathbb{I} \\ \mathrm{i}\mathbb{I} & \mathbb{I} \end{bmatrix}$. However, in our case the matrix L is particularly structured and such an example is unlikely. We leave the further theoretical analysis as an open problem and point to the work of Higham [35] that provides additional insight into the structure of completely pivoted LU decomposition of complex symmetric matrices.[8]

5.1.3. Line 5. As a consequence of the results in Section 5.1.1 and Section 5.1.2, the matrix $B = U^T Y_A^T D_A$ satisfies

$$\min_{\Delta=\mathrm{diag}} \kappa_2(B\Delta) \le \kappa_2(U)\kappa_2(Y_A),$$
$$\kappa_2(B_c) \le \sqrt{n}\kappa_2(U)\kappa_2(Y_A), \qquad \text{where } B_c = B\mathrm{diag}(1/\|B(:,i)\|_2).$$

The upper triangular matrix R computed by the pivoted QR factorization $B(:,\pi_5) = QR$ in step 5 also deserves our attention. As a consequence of pivoting [9], the matrix R satisfies

$$|R_{ii}| \ge \sqrt{\sum_{k=i}^{j} |R_{kj}|^2}, \qquad \text{for all } 1 \le i \le j \le n. \tag{5.1}$$

[8] Higham considered complex symmetric matrices with positive definite real and imaginary parts—a structure that our matrix Z does not have.

ETNA
Kent State University
http://etna.math.kent.edu

Although originally devised to reveal the numerical rank, in our setting the pivoting and the resulting structure (5.1) are understood and exploited as strong preconditioners. Let $\Delta_r = \mathrm{diag}(1/\|R(i,:)\|_2)_{i=1}^n$, $R_r = \Delta_r R$, $\Delta_c = \mathrm{diag}(1/\|R(:,i)\|_2)_{i=1}^n$, and $R_c = R\Delta_c$. Then it can be shown that $\kappa_2(R_r) \le n^{3/2}\kappa_2(R_c) = n^{3/2}\kappa_2(B_c)$. We can easily understand this mechanism if we note that (5.1) implies

$$|(R_r^{-1})_{ij}| \le \sqrt{n-j+1}\left|\frac{R_{jj}}{R_{ii}}\right| |(R_c^{-1})_{ij}|, \quad \text{i.e.,} \quad |R_r^{-1}| \le \sqrt{n}|R_c^{-1}|,$$

where in the case of an ill-conditioned matrix B, it is possible that $\kappa_2(R_r) \ll \kappa_2(R_c)$. In our concrete example, $\kappa_2(R_r) \approxeq$ 2.5989e+000, $\kappa_2(R_c) = \kappa_2(B_c) \approxeq$ 1.6051e+002, and with $\Delta_d = \mathrm{diag}(1/|R_{ii}|)$, we have $\kappa_2(\Delta_d) \approxeq$ 6.2400e+258 and $\kappa_2(\Delta_d R) \approxeq$ 2.6852e+000. Compare this with $\kappa_2(B) > 10^{258}$.

5.1.4. Line 6. Finally, the matrix $S = U^T X_A(:,\pi_5)R^T = U^T X_A(:,\pi_5)R_r^T\Delta_r^{-1}$ satisfies

$$\begin{aligned} \kappa_2(S\Delta_s) &\le \sqrt{n}\kappa_2(U^T X_A(:,\pi_5)R_r^T) \le \sqrt{n}\kappa_2(U)\kappa_2(X_A)\kappa_2(R_r), \\ \Delta_s &= \mathrm{diag}(1/\|S(:,i)\|_2). \end{aligned} \tag{5.2}$$

In our example, $\kappa_2(S\Delta_s) \approxeq$ 1.4481e+002 and $\kappa_2(S) > 10^{260}$. In such a situation, the Jacobi SVD algorithm can compute the SVD decomposition of S to nearly machine precision.

REMARK 5.1. Before going to the error analysis, note that in Example 4.2 the condition numbers of column- or row-scaled matrices that we pointed out were at most of the order of 10^3, which is in perfect accordance with the measured errors in the singular values and in the singular vectors reported in Figure 4.2. The key of the accuracy of the proposed algorithms that will become apparent in Section 5.2 is that at no point they generate floating-point errors that can trigger high condition numbers of $\mathcal{V}(x)$, $\mathcal{C}$, A, B, or S.

REMARK 5.2. The condition number is a function of a matrix that has its own condition number, and one should be careful when computing and using computed condition numbers. In fact, up to certain multiplicative constants, the condition number of the condition number is the condition number [12, 34]. So, for instance, in Matlab `cond(hilb(100))` gives the answer `6.9789e+19`, while the true condition number is above 10^{150}. This severe underestimate can be partially understood by combining the backward error framework of the numerical SVD (used by `cond()`) and the analysis of the condition number under random perturbations [43, 46]. The condition numbers we reported in this section are computed accurately enough to give the true estimate of the sensitivity.

5.2. Error analysis. The floating-point error analysis of an elaborate numerical algorithm is usually technically involved, but at some point it has to be done. Such an analysis, combined with the corresponding perturbation theory, is the only way to provide mathematically rigorous statements about the numerical properties of an algorithm. We follow the lines of Algorithm 3 and use the well-established and understood error bounds for the basic operations such as matrix multiplication, LU decomposition, QR factorization, and the Jacobi SVD computation as outlined in Section 2.2 but without going into the unnecessary technical details. Instead, we identify the key moments and discuss how we contrived to curb the ill-conditioning throughout the algorithm. We use $\mathrm{fl}(\mathrm{expression})$ to denote the value of expression computed in floating-point arithmetic. $\widehat{\varepsilon}$ is the machine roundoff unit. Occasionally, we use the following notation: for an entry-wise nonnegative matrix B, we denote a matrix A by $\lceil B \rceil$, or write $A \in \lceil B \rceil$ if, entry-wise, $|A| \le B$.

ETNA
Kent State University
http://etna.math.kent.edu

5.2.1. Line 2. To ease the notation, assume that the input is so ordered that the permutations π_1, π_2 are identities. Since the LDU of the scaled Cauchy matrix $\sqrt{D_3}\mathcal{C}$ in line 2 is computed in an entry-wise forward stable way, the computed matrices $\widetilde{L} = L + \delta L$, $\widetilde{U} = U + \delta U$, $\widetilde{D}_4 = D_4 + \delta D_4$ satisfy

$$|\delta L| \le \epsilon_1 |L|, \quad |\delta U| \le \epsilon_1 |U|, \quad |\delta D_4| \le \epsilon_1 |D_4|, \qquad \text{where } \epsilon_1 \le O(n)\widehat{\varepsilon}. \tag{5.3}$$

We stress that this computation is rather subtle and that in extreme cases it may require the complex roots of unity tabulated in triple precision. The reader is referred to [13, Section 5] for all the fine and nontrivial details.

5.2.2. Line 3. The effect of δL, δU, and δD_4 on the computed values of the matrix $A = D_4 L^T L D_4$ is estimated next. The exact product $\check{A} = \widetilde{D}_4 \widetilde{L}^T \widetilde{L} \widetilde{D}_4$ of the computed factors satisfies

$$\begin{aligned} &\check{A} = D_4(\mathbb{I} + \delta D_4 D_4^{-1})\widetilde{L}^T \widetilde{L}(\mathbb{I} + \delta D_4 D_4^{-1})D_4 = D_4 \check{L}^T \check{L} D_4, \\ &\check{L} = L + \Delta L, \quad |\Delta L| \le (2\epsilon_1 + \epsilon_1^2)|L|. \end{aligned} \tag{5.4}$$

But $\check{A}$ must be computed in floating-point arithmetic. The resulting matrix $\widetilde{A} = \mathfrak{fl}(\widetilde{D}_4 \widetilde{L}^T \widetilde{L} \widetilde{D}_4)$ is obtained in two steps. First, apply a diagonal scaling with $\widetilde{D}_4 = D_4 + \lceil \epsilon_1 D_4 \rceil$ to get

$$\begin{aligned} \mathfrak{fl}(\widetilde{L}\widetilde{D}_4) &= (L + \lceil \epsilon_1 |L| \rceil + \lceil \widehat{\varepsilon} |L + \lceil \epsilon_1 |L| \rceil| \rceil)(\mathbb{I} + \lceil \epsilon_1 \mathbb{I} \rceil) D_4 \\ &= (L + \widetilde{\delta L}) D_4, \qquad \widetilde{\delta L} \in \lceil \eta_L |L| \rceil, \quad \eta_L \le \epsilon_1(2 + 2\widehat{\varepsilon} + \epsilon_1) + \widehat{\varepsilon}(1 + \epsilon_1^2), \end{aligned}$$

and then compute the cross-product $\widetilde{A} = A + \delta A$ as

$$\begin{aligned} \widetilde{A} &= \mathfrak{fl}(\mathfrak{fl}(\widetilde{L}\widetilde{D}_4)^T \mathfrak{fl}(\widetilde{L}\widetilde{D}_4)) \\ &= D_4(L + \widetilde{\delta L})^T (L + \widetilde{\delta L}) D_4 + \left\lceil \epsilon_2 |D_4| |L + \widetilde{\delta L}|^T |L + \widetilde{\delta L}| |D_4| \right\rceil \\ &= D_4 \left((L + \widetilde{\delta L})^T (L + \widetilde{\delta L}) + \left\lceil \epsilon_2 |L + \widetilde{\delta L}|^T |L + \widetilde{\delta L}| \right\rceil \right) D_4 = D_4(Z + \delta Z) D_4, \end{aligned} \tag{5.5}$$

where

$$|\delta Z| \le \theta |L|^T |L|, \quad \theta \le ((2\eta_L + \eta_L^2)(1 + \epsilon_2) + \epsilon_2), \quad \epsilon_2 \le O(n)\widehat{\varepsilon}, \quad \text{and} \tag{5.6}$$

$$\|\delta Z\|_2 \le \zeta \|Z\|_2, \quad \text{with } \zeta = \theta \| |L^{-1}| |L^{-T}| |L|^T |L| \|_2 \le \theta n^2 \kappa_2(L)^2. \tag{5.7}$$

We stress that the matrices Z and $\widetilde{Z} = Z + \delta Z$ are introduced here only for the purpose of the analysis. They are not computed in the algorithm, that is, A and $\widetilde{A}$ are not explicitly scaled with D_4^{-1}.

5.2.3. Line 4. This is the defining moment of the algorithm. Instead of the matrix $A = D_4 Z D_4$, we have an explicitly computed $\widetilde{A} = D_4(Z + \delta Z)D_4$ with a small matrix δZ that can be estimated using (5.6), (5.7). The key question is whether $\widetilde{A}$ contains accurate information on the SVD of A. If $Z = L_Z U_Z$ is the LU decomposition of Z, then

$$\begin{aligned} \widetilde{A} &= D_4(L_Z U_Z + \delta Z) D_4 = D_4 L_Z (\mathbb{I} + L_Z^{-1} \delta Z U_Z^{-1}) U_Z D_4 \\ &= D_4 L_Z \widehat{L}\widehat{U} U_Z D_4 = D_4 L_Z \widehat{L} L_Z^{-1} D_4^{-1} A D_4^{-1} U_Z^{-1} \widehat{U} U_Z D_4, \end{aligned} \tag{5.8}$$

where $\mathbb{I} + L_Z^{-1} \delta Z U_Z^{-1} = \widehat{L}\widehat{U}$ is the LU decomposition of a perturbation of the identity. If we set $E_Z = L_Z^{-1} \delta Z U_Z^{-1}$, then, using (5.6), $\|E_Z\|_F \le \theta \|L_Z^{-1}\|_2 \|U_Z^{-1}\|_2 \|L\|_F^2$. If $\|E_Z\|_F < 1$,

ETNA
Kent State University
http://etna.math.kent.edu

then we can apply a perturbation result from [5] and write $\widehat{L} = \mathbb{I} + E_L$, $\widehat{U} = \mathbb{I} + E_U$ with a strictly lower triangular matrix E_L, $\|E_L\|_F \leq \|E_Z\|_F/(1 - \|E_Z\|_F)$ and an upper triangular one E_U, $\|E_U\|_F \leq \|E_Z\|_F/(1 - \|E_Z\|_F)$. Hence,

$$\widetilde{A} = (\mathbb{I} + D_4 L_Z E_L L_Z^{-1} D_4^{-1}) A (\mathbb{I} + D_4^{-1} U_Z^{-1} E_U U_Z D_4) \equiv (\mathbb{I} + \Xi_1) A (\mathbb{I} + \Xi_2). \tag{5.9}$$

A careful reading of (5.9) reveals that the diagonal matrix D_4, no matter how ill-conditioned, cannot inflate the error terms provided that the diagonal entries of D_4 are monotonically decreasing in modulus or if the ratio $\tau = \max_{j>i} |(D_4)_{jj}|/|(D_4)_{ii}|$ is not too much larger than one. This is due to the structure of the products $D_4(\textit{lower triangular})D_4^{-1}$ and $D_4^{-1}(\textit{upper triangular})D_4$. Interestingly, if $\tau < 1$, then the ill-conditioning revealed in D_4 actually curbs the error. At this point, it is instructive to recall Figure 5.1.

REMARK 5.3. Note that in this analysis we can even assume that, on entry to these formulas, the matrix $\widetilde{A}$ in (5.8) has been permuted according to a complete pivoting strategy for its LU decomposition. If π_3, π_4 are the actually computed permutations, we can repeat the above with A, $\widetilde{A}$, Z, δZ, D_4 replaced with, respectively, $A(\pi_3, \pi_4)$, $\widetilde{A}(\pi_3, \pi_4)$, $Z(\pi_3, \pi_4)$, $\delta Z(\pi_3, \pi_4)$, $D_4(\pi_3, \pi_3)$, or $D_4(\pi_4, \pi_4)$. Then, an analogous relation to (5.9) holds with these permuted matrices and appropriately redefined matrices L_Z, U_Z, E_L, and E_U. If the complete pivoting performs well, then the value of τ will be below or at least not much larger than one.

Now, an application of [25, Theorem 3.1] to (5.9) (cf. (2.3), (2.4)) yields the key factors that determine the accuracy of the SVD,[9]

$$\begin{aligned} \xi_1 = \|\Xi_1\|_2 \equiv \|D_4 L_Z E_L L_Z^{-1} D_4^{-1}\|_2 &\leq \tau \kappa_2(L_Z) \|E_L\|_F \\ &\leq \tau \kappa_2(L_Z) \|E_Z\|_F/(1 - \|E_Z\|_F), \end{aligned} \tag{5.10}$$

$$\begin{aligned} \xi_2 = \|\Xi_2\|_2 \equiv \|D_4^{-1} U_Z^{-1} E_U U_Z D_4\|_2 &\leq \tau \kappa_2(U_Z) \|E_U\|_F \\ &\leq \tau \kappa_2(U_Z) \|E_Z\|_F/(1 - \|E_Z\|_F). \end{aligned} \tag{5.11}$$

Hence, if Z allows for an accurate LU decomposition under the perturbation δZ, then ξ_1, ξ_2 are small and an accurate approximation of the SVD of A is contained in $\widetilde{A}$ with each singular value determined up to a relative error of order of $\xi_1 + \xi_2$ at most. How to extract such an approximation is another question.

To that end, consider the LU decomposition with complete pivoting of $\widetilde{A} = D_4 \widetilde{Z} D_4$, the one that is actually computed in the algorithm. To ease the notation, we temporarily denote the pivoted matrix $\widetilde{A}(\pi_3, \pi_4)$ with $\widehat{A}$ and write $\widehat{A} = \widehat{D_4} \widehat{Z} \widehat{D_4}$, where $\widehat{D_4}$ is the pivoted diagonal matrix D_4 and $\widehat{Z}$ is the corresponding permuted matrix $\widetilde{Z} = Z + \delta Z$. The computed LU decomposition $\widetilde{L}_{\widehat{A}} \widetilde{U}_{\widehat{A}}$ of $\widehat{A}$ satisfies the residual (backward error) relation $\widehat{A} + \delta\widehat{A} = \widetilde{L}_{\widehat{A}} \widetilde{U}_{\widehat{A}}$, where the backward error satisfies $|\delta\widehat{A}| \leq \epsilon_{LU} |\widetilde{L}_{\widehat{A}}| |\widetilde{U}_{\widehat{A}}|$, $\epsilon_{LU} \leq O(n)\widehat{\epsilon}$. With the uniqueness theorem for the LU decomposition in mind, write

$$\widehat{A} + \delta\widehat{A} = \widehat{D_4}(\widehat{Z} + \delta\widehat{Z})\widehat{D_4}, \qquad \text{where } \delta\widehat{Z} = \widehat{D_4}^{-1} \delta\widehat{A} \widehat{D_4}^{-1},$$

to conclude that $\widehat{Z} + \delta\widehat{Z} = \left(\widehat{D_4}^{-1} \widetilde{L}_{\widehat{A}} \widehat{D_4}\right) \left(\widehat{D_4}^{-1} \widetilde{U}_{\widehat{A}} \widehat{D_4}^{-1}\right) \equiv \widetilde{L}_{\widehat{Z}} \widetilde{U}_{\widehat{Z}}$ is the LU decomposition of $\widehat{Z} + \delta\widehat{Z}$. Note that here $|\delta\widehat{Z}| \leq \epsilon_{LU} |\widetilde{L}_{\widehat{Z}}| |\widetilde{U}_{\widehat{Z}}|$. Now write

$$\widehat{Z} = \widetilde{L}_{\widehat{Z}} \widetilde{U}_{\widehat{Z}} - \delta\widehat{Z} = \widetilde{L}_{\widehat{Z}} (\mathbb{I} - \widetilde{L}_{\widehat{Z}}^{-1} \delta\widehat{Z} \widetilde{U}_{\widehat{Z}}^{-1}) \widetilde{U}_{\widehat{Z}} = \widetilde{L}_{\widehat{Z}} \breve{L} \breve{U} \widetilde{U}_{\widehat{Z}},$$

[9] Here we carefully combine the properties of the spectral and the Frobenius norms.

ETNA
Kent State University
http://etna.math.kent.edu

where $\breve{L}\breve{U}$ is the LU decomposition of a perturbation of the identity, $\breve{L} = \mathbb{I} + E_{\breve{L}}$, $\breve{U} = \mathbb{I} + E_{\breve{U}}$. Hence,

$$\begin{aligned}(5.12)\quad \widehat{A} &= \widehat{D_4}\widetilde{L}_{\widehat{Z}}\breve{L}\widetilde{L}_{\widehat{Z}}^{-1}\widehat{D_4}^{-1}(\widehat{A}+\delta\widehat{A})\widehat{D_4}^{-1}\widetilde{U}_{\widehat{Z}}^{-1}\breve{U}\widetilde{U}_{\widehat{Z}}\widehat{D_4}\\ &= (\mathbb{I}+\widehat{D_4}\widetilde{L}_{\widehat{Z}}E_{\breve{L}}\widetilde{L}_{\widehat{Z}}^{-1}\widehat{D_4}^{-1})(\widetilde{L}_{\widehat{A}}\widetilde{U}_{\widehat{A}})(\mathbb{I}+\widehat{D_4}^{-1}\widetilde{U}_{\widehat{Z}}^{-1}E_{\breve{U}}\widetilde{U}_{\widehat{Z}}\widehat{D_4})\\ (5.13)\quad &\equiv (\mathbb{I}+\widehat{\Xi}_1)\widetilde{L}_{\widehat{A}}\widetilde{U}_{\widehat{A}}(\mathbb{I}+\widehat{\Xi}_2) = (\mathbb{I}+\Xi_1)A(\pi_3,\pi_4)(\mathbb{I}+\Xi_2)\quad \text{(using Remark 5.3)}.\end{aligned}$$

It is immediate that the values of $\widehat{\xi}_1 = \|\widehat{D_4}\widetilde{L}_{\widehat{Z}}E_{\breve{L}}\widetilde{L}_{\widehat{Z}}^{-1}\widehat{D_4}^{-1}\|_2$ and $\widehat{\xi}_2 = \|\widehat{D_4}^{-1}\widetilde{U}_{\widehat{Z}}^{-1}E_{\breve{U}}\widetilde{U}_{\widehat{Z}}\widehat{D_4}\|_2$ can be bounded analogously to ξ_1 and ξ_2 in (5.10) and (5.11). If these four numbers are small, then the computed pivoted LU decomposition of $\widetilde{A}$ is accurate. Furthermore, if $A(\pi_3,\pi_4) = L_A U_A$ is the exact LU decomposition and if we repeat the analysis in (5.8)–(5.9) with an appropriately permuted A and $\widetilde{A}$ as explained in Remark 5.3, then by the triangular structure of the multiplicative perturbations and by the uniqueness of the LU decomposition,

$$\widetilde{L}_{\widehat{A}} = (\mathbb{I}+\widehat{\Xi}_1)^{-1}(\mathbb{I}+\Xi_1)L_A = (\mathbb{I}+\Xi_3)L_A,\quad \widetilde{U}_{\widehat{A}} = U_A(\mathbb{I}+\Xi_2)(\mathbb{I}+\widehat{\Xi}_2)^{-1} = U_A(\mathbb{I}+\Xi_4).$$

This implies

$$\frac{\|\widetilde{L}_{\widehat{A}}-L_A\|_2}{\|L_A\|_2} \le O(\xi_1+\widehat{\xi}_1),\qquad \max_{i=1:n}\frac{|(\widetilde{U}_{\widehat{A}})_{ii}-(U_A)_{ii}|}{|(U_A)_{ii}|} \le O(\xi_2+\widehat{\xi}_2),$$
$$\frac{\|\widetilde{U}_{\widehat{A}}-U_A\|_2}{\|U_A\|_2} \le O(\xi_2+\widehat{\xi}_2).$$

Let now $\widetilde{D}_A = \mathrm{diag}(\widetilde{U}_{\widehat{A}})$, and consider the scaling $\mathfrak{fl}(\widetilde{D}_A^{-1}\widetilde{U}_{\widehat{A}})$.

$$\begin{aligned}\widetilde{T}_A &\equiv \mathfrak{fl}(\widetilde{D}_A^{-1}\widetilde{U}_{\widehat{A}}) = \widetilde{D}_A^{-1}(\widetilde{U}_{\widehat{A}} + \left\lceil \widehat{\varepsilon}|\widetilde{U}_{\widehat{A}}|\right\rceil)\\ &= (\mathbb{I}+F)D_A^{-1}(U_A + U_A\Xi_4 + \lceil\widehat{\varepsilon}|U_A+U_A\Xi_4|\rceil)\\ (5.14)\quad &= (\mathbb{I}+F)(T_A + T_A\Xi_4 + \lceil\widehat{\varepsilon}(|T_A|+|T_A\Xi_4|)\rceil = T_A+\delta T_A,\quad T_A = D_A^{-1}U_A,\end{aligned}$$

where the terms that form δT_A are T_A multiplied from the right by an error matrix of small norm or/and scaled from the left by the small diagonal error matrix F or/and multiplied by the roundoff $\widehat{\varepsilon}$. From this we conclude that $\|\delta T_A(i,:)\|_2 \le \vartheta\|T_A(i,:)\|_2$ for all i, where $0 \le \vartheta \le O(\|\Xi_4\|_2) + O(\xi_2+\widehat{\xi}_2) + O(\widehat{\varepsilon})$. If we write these decompositions in the form of the RRDs, we have $A = (\Pi_3 L_A)D_A(T_A\Pi_4^T) \equiv X_A D_A Y_A^*$ with $X_A = \Pi_3 L_A$, $Y_A = \Pi_4 T_A^*$. For the computed decomposition, we have $\widetilde{A}+\delta\widetilde{A} = \widetilde{X}_A\widetilde{D}_A\widetilde{Y}_A^*$ with $\widetilde{X}_A = \Pi_3\widetilde{L}_{\widehat{A}}$ and $\widetilde{Y}_A = \Pi_4\widetilde{T}_A^*$. The important thing is now that in the next step we use the computed decomposition

$$\begin{aligned}\widetilde{A}+\delta\widetilde{A} = \widetilde{X}_A\widetilde{D}_A\widetilde{Y}_A^* &= (X_A+\delta X_A)(D_A+\delta D_A)(Y_A+\delta Y_A)^*\\ &= (X_A+\delta X_A)D_A(Y_A+\Delta Y_A)^*,\end{aligned}$$

where $\delta\widetilde{A}$ is the backward error and the errors in the factors (as compared to the corresponding exact RRD of the exact matrix A) are bounded by

$$(5.15)\qquad \frac{\|\delta X_A\|_2}{\|X_A\|_2} \le \xi_A,\quad \frac{\|\delta Y_A\|_2}{\|Y_A\|_2} \le \eta_A,\quad \frac{|(\delta D_A)_{ii}|}{|(D_A)_{ii}|} \le \delta_A.$$

From the previous analysis, it is evident that

$$\xi_A \le O(\xi_1+\widehat{\xi}_1),\quad \eta_A \le O(\|\Xi_4\|_2) + O(\xi_2+\widehat{\xi}_2) + O(\widehat{\varepsilon}),\quad \text{and } \delta_A \le O(\xi_1+\widehat{\xi}_1).$$

ETNA
Kent State University
http://etna.math.kent.edu

Note that we have avoided making any statement about the backward error $\delta\widetilde{A}$. Instead, the multiplicative form of the perturbation (5.12)–(5.13) has been used to derive the perturbation estimate (5.15) for the RRD of A. (Also, we could have taken $\widetilde{D}_A = \mathbb{I}$ and $\widetilde{Y}_A = \Pi_4\widetilde{U}_{\widetilde{A}}^*$, which would then require some minor technical changes in Section 5.2.4 below without significantly changing the final estimates.)

We should point out that an RRD can be computed by other means, for instance, by using the SVD of $\widetilde{A}$ and thus giving other bounds for (5.15). In that case the condition number of $\widetilde{Y}_A$ is approximately one, which improves the theoretical error bound in Section 5.2.4. However, it seems that such an extra computational effort is not necessary in practice as the completely pivoted LDU performs very well; cf. the last paragraph in Section 5.1.1.

5.2.4. Line 5. On input to line 5, the data are $U + \delta U$, $Y_A + \delta Y_A$, and $D_A + \delta D_A$. Consider first

$$\widetilde{\Upsilon}_A \equiv \mathrm{fl}((Y_A^* + \delta Y_A^*)^T(D_A + \delta D_A)) = (Y_A^* + \widetilde{\delta Y_A^*})^T D_A,$$

where $\|\widetilde{\delta Y_A}\|_2 \le \widetilde{\eta_A}\|Y_A\|_2$ and $\widetilde{\eta_A} \le \eta_A + (\delta_A + \widehat{\varepsilon}\sqrt{n}(1+\delta_A))(1+\eta_A)$. In the next step, we have

$$\begin{aligned}
\widetilde{B} &= \mathrm{fl}(\widetilde{U}^T\widetilde{\Upsilon}_A) = \widetilde{U}^T\widetilde{\Upsilon}_A + \left\lceil O(n)\widehat{\varepsilon}|\widetilde{U}^T||\widetilde{\Upsilon}_A|\right\rceil \\
&= (\mathbb{I} + \underbrace{\delta U^T U^{-T}}_{\Omega_1}) \\
&\quad \left(\mathbb{I} + \underbrace{U^T\widetilde{\delta Y_A^*}^T Y_A^{-*T}U^{-T} + U^T\widetilde{U}^{-T}\left\lceil O(n)\widehat{\varepsilon}|\widetilde{U}^T||\widetilde{\Upsilon}_A|\right\rceil D_A^{-1}Y_A^{-*T}U^{-T}}_{\Omega_2}\right) U^T Y_A^{*T} D_A \\
&= (\mathbb{I} + \Omega_B)B, \qquad \text{where } \Omega_B = \Omega_1 + \Omega_2 + \Omega_1\Omega_2 \quad \text{and}
\end{aligned}$$

$$\begin{aligned}
(5.16)\quad \|\Omega_1\|_2 &= \|U^{-1}\delta U\|_2 \le \kappa_2(U)\frac{\|\delta U\|_2}{\|U\|_2} \qquad (\text{or, e.g., } \|\Omega_1\|_2 \le \epsilon\||U^{-1}||U|\|_2), \\
\|\Omega_2\|_2 &\le \kappa_2(U)\kappa_2(Y_A)(\widetilde{\eta}_A + O(n^2)\widehat{\varepsilon}(1 + \|\widetilde{U}^{-1}\delta U\|_2)(1 + \frac{\|\delta U\|_2}{\|U\|_2})(1+\widetilde{\eta}_A)).
\end{aligned}$$

Let now $\widetilde{B}\pi_5 \approx \widetilde{Q}\widetilde{R}$ be the computed QR factorization. By the backward error analysis, there exist a matrix $\delta\widetilde{B}$ and a unitary matrix $\widehat{Q} \approx \widetilde{Q}$ such that $(\widetilde{B} + \delta\widetilde{B})\pi_5 = \widehat{Q}\widetilde{R}$ and $\|\delta\widetilde{B}(:,i)\|_2 \le \beta_{QR}\|\widetilde{B}(:,i)\|_2$ with $\beta_{QR} \le O(n)\widehat{\varepsilon}$ for all i. Thus, with some ΔB,

$$\begin{aligned}
(\widetilde{B} + \delta\widetilde{B})\pi_5 \equiv (B + \Delta B)\pi_5 = \widehat{Q}\widetilde{R}, \qquad \|\Delta B(:,i)\|_2 \le \eta_B\|B(:,i)\|_2, \\
\eta_B = \|\Omega_B\|_2 + \beta_{QR}(1 + \|\Omega_B\|_2).
\end{aligned}$$

The permutation π_5 ensures that $\widetilde{R}$ has the structure (5.1). Next, we need to know how much the computed $\widetilde{R}$ differs from the true triangular factor $R = \widetilde{R} + \delta\widetilde{R}$. For our targeted high accuracy, the standard bound on $\|\delta\widetilde{R}\|_2/\|R\|_2$ is not good enough. We need more structured information, and we use the results of [22, Section 6.1], which states that $\delta\widetilde{R} = \Gamma\widetilde{R}$, where Γ is upper triangular with the Frobenius norm bound

$$\|\Gamma\|_F \le \frac{\sqrt{8n}\eta_B\|\widetilde{R}_c^{-1}\|_2}{1-\eta_B} + \sqrt{2}\frac{n(\eta_B\|\widetilde{R}_c^{-1}\|_2)^2}{(1-\eta_B)^2}, \qquad \widetilde{R}_c = \widetilde{R}\mathrm{diag}(1/\|\widetilde{R}(:,i)\|_2)_{i=1}^n.$$

ETNA
Kent State University
http://etna.math.kent.edu

From this, we not only have multiplicative perturbations $R = (\mathbb{I}+\Gamma)\widetilde{R}$, with $\widetilde{R} = (\mathbb{I}+\Gamma)^{-1}R \approx (\mathbb{I}-\Gamma)R$, but also small column-wise error bounds $\|\delta\widetilde{R}(:,i)\|_2 \leq \|\Gamma\|_2\|\widetilde{R}(:,i)\|_2$ and also favorable row-wise bounds[10] $\|\delta\widetilde{R}(i,:)\|_\infty \leq \|\Gamma(i,:)\|_2\|\widetilde{R}(i,:)\|_\infty$. For our purposes, we rewrite this as $\widetilde{R} = R + \delta R$ with

$$\begin{aligned} \|\delta R(i,:)\|_\infty &\leq \frac{\|\Gamma(i,:)\|_2}{1-\|\Gamma(i,:)\|_2}\|R(i,:)\|_\infty, \qquad \text{and thus} \\ \|\delta R(i,:)\|_2 &\leq \sqrt{n}\gamma\|R(i,:)\|_2, \qquad\qquad i = 1,\dots n, \end{aligned} \tag{5.17}$$

where $\gamma = \max_i \frac{\|\Gamma(i,:)\|_2}{1-\|\Gamma(i,:)\|_2}$. Here we use the reasonable assumption that $\|\Gamma\|_F < 1$. (See the condition numbers in Section 5.1.3.)

5.2.5. Line 6. Let $\widetilde{S} = \mathfrak{fl}(\mathfrak{fl}(\widetilde{U}^T\widetilde{X}_A)\widetilde{R}^T)$ be the computed value of S. We have

$$\begin{aligned} \widetilde{S} &= \widetilde{U}^T\widetilde{X}_A\widetilde{R}^T + \left[\epsilon_1|\widetilde{U}^T||\widetilde{X}_A|\right]\widetilde{R}^T + \left[\epsilon_2\left|\widetilde{U}^T\widetilde{X}_A + \left[\epsilon_1|\widetilde{U}^T||\widetilde{X}_A|\right]\right||\widetilde{R}^T|\right] \\ &= (\mathbb{I}+\delta U^T U^{-T})(\mathbb{I}+U^T\delta X_A X_A^{-1}U^{-T})(\mathbb{I}+U^T X_A\delta R^T R^{-T}X_A^{-1}U^{-T})U^T X_A R^T \\ &\quad + \left[\epsilon|\widetilde{U}^T||\widetilde{X}_A||\widetilde{R}^T|\right](R^{-T}X_A^{-1}U^{-T})U^T X_A R^T \\ &= ((\mathbb{I}+\Psi_1)(\mathbb{I}+\Psi_2)(\mathbb{I}+\Psi_3)+\Psi_4)\,S = (I+\Psi_S)S, \end{aligned}$$

where $S = U^T X_A R^T$, $\epsilon = \epsilon_1+\epsilon_2+\epsilon_1\epsilon_2$, and $\epsilon_1, \epsilon_2 \leq O(n)\widehat{\varepsilon}$. Further, $\Psi_1 = \Omega_1$ is bounded as in (5.16), and for the remaining Ψ_i's, we use Section 5.1.3 and (5.17) to obtain

$$\begin{aligned} &\|\Psi_2\|_2 \leq \kappa_2(U)\kappa_2(X_A)\xi_A, \\ &\|\Psi_3\|_2 \leq \kappa_2(U)\kappa_2(X_A)\|R^{-1}\delta R\|_2 \leq n\kappa_2(U)\kappa_2(X_A)\|R_r^{-1}\|_2\gamma, \\ &\|\Psi_4\|_2 \leq \left\||\widetilde{U^T}||\widetilde{X}_A||\widetilde{R}^T\Delta_r^{-1}|\right\|_2 \left\|R_r^{-1}X_A^{-1}U^{-T}\right\|_2 \epsilon \lesssim n^{3/2}\kappa_2(U)\kappa_2(X_A)\kappa_2(R_r)\epsilon, \end{aligned}$$

where the last inequality holds up to a factor of

$$(1+\|U^{-1}\delta U\|_2)(1+\|\delta X_A X_A^{-1}\|_2)(1+\|R^{-1}\delta R\|_2).$$

5.2.6. Line 7. Finally, the Jacobi SVD computes the SVD of $\widetilde{S} = S + \delta S$ with $\delta S = \Psi_S S$. We know from Section 2.2 that this SVD corresponds to the exact SVD of $\widetilde{S}+\Delta\widetilde{S}$, where $\|\Delta\widetilde{S}(:,i)\|_2 \leq \xi_J\|\widetilde{S}(:,i)\|_2$. Altogether, we have computed the SVD $\widehat{U}_s\widetilde{\Sigma}\widehat{W}_s^*$ of $S+\Delta S \equiv S + (\delta S + \Delta\widetilde{S})$, i.e.,

$$\begin{aligned} S+\Delta S = \widehat{U}_s\widetilde{\Sigma}\widehat{W}_s^*, \qquad &\|\Delta S(:,i)\|_2 \leq \zeta_S\|S(:,i)\|_2, \quad i = 1,\dots,n; \\ &\zeta_S \leq \|\Psi_S\|_2 + \xi_J(1+\|\Psi_S\|_2). \end{aligned}$$

Finally, calling to the discussion in Section 2 (in particular (2.1)), Section 5.1, and the condition number estimate (5.2), we obtain a theoretical forward error bound and a match with the measured errors in numerical experiments. For a detailed statement on the accuracy of the computed singular values and vectors we refer to [23, Section 5.4, Section 5.5].

[10] Note that $|\delta\widetilde{R}_{ij}| = |\sum_{k=i}^{j}\Gamma_{ik}\widetilde{R}_{kj}| \leq \|\Gamma(i,:)\|_2\|\widetilde{R}(i:j,j)\|_2 \leq \|\Gamma(i,:)\|_2|\widetilde{R}_{ii}|$, where the last inequality follows from (5.1).

ETNA
Kent State University
http://etna.math.kent.edu

5.3. Perturbation theory. The preceding analysis provides an error bound for the computed SVD that corresponds to the input vectors $x, d \in \mathbb{C}^n$ stored in computer memory in working-precision format. We can use it to algorithmically derive a perturbation estimate, i.e., to bound the sensitivity of the SVD to the changes $x \rightsquigarrow \widetilde{x} = x + \delta x$, $d \rightsquigarrow \widetilde{d} = d + \delta d$. Note that the accuracy of the computed SVD and the sensitivity of the SVD to the noise in the input data are fundamentally different issues. In the case of ill-conditioned matrices, it is possible that we can compute to full machine precision the singular values of the matrix that corresponds to the stored parameters and that at the same time those values are highly sensitive even to the tiniest changes in the parameters. The key to estimate this sensitivity is (5.3), where the value of ϵ_1 can be bounded in terms of $\max_i |\delta x_i/x_i|$, $\max_i |\delta d_i/d_i|$, the relative gap between the x_i's, and the roots of unity similarly as in [16, Theorem 5.1]. Then we follow the steps of Section 5.2 but ignore the floating-point errors (i.e., set $\widehat{\varepsilon} = 0$) and let only ϵ_1 propagate throughout the algorithm. So, for instance, we use (5.4) instead of (5.5) and in the final two steps $\Psi_4 = 0$, $\xi_J = 0$. We leave the details out for the sake of brevity.

5.4. Robust computations in the full range of machine numbers. Since the condition number of $\mathcal{H} \in \mathbb{C}^{n\times n}$ grows exponentially with n, a robust software implementation of the new algorithms faces more problems related to finite precision: the singular values spread very quickly over the range of positive machine numbers and reach its boundaries, the underflow and overflow thresholds. If we use Matlab with IEEE double-precision floating-point arithmetic, then the underflow threshold is $\alpha = 2.225073858507201\text{e}{-308}$, and the overflow threshold is $\omega = 1.797693134862316\text{e}{+308}$. Numerical procedures for computing the SVD such as, e.g., `XGESVD()`, `XGESDD()` from LAPACK [2], are equipped with a safeguard mechanism to prevent overflow that scales the matrix so that during the computation, the largest singular value remains below $\sqrt{\omega}$ and on exit, the computed singular values are rescaled. In the case of an extremely large quotient σ_1/σ_n, this scaling may irreparably damage the information on the smallest singular values. A particularly interesting illustration of the robustness of our software implementation is that it computes the singular values to high relative accuracy in the full range $[\alpha, \omega]$ of the working precision without ever having to resort to higher precision. We will not go into all technical details because they can be rather involved. Instead, we only illustrate one, and we provide references to detailed analyses.

Assume that the matrix S in line 6 of Algorithm 3 is computed without underflow/overflow, and consider computing its SVD. The problem here is that the Jacobi SVD algorithm is implicit and the Jacobi rotations are designed to orthogonalize pivot columns. In the extreme cases, the tangent of the Jacobi angle underflows—it either gets denormalized or flushed to zero, meaning that the rotation is inaccurate and without orthogonalizing effect. Also, the Gram matrix of the pair of pivot columns, used to determine the rotation, may not have a finite representation in working precision. Other methods such as the QR decomposition are also helpless because they also implicitly work on S^*S. This is easily illustrated (and explained in detail, see [24]) in Matlab:

$$A = \begin{bmatrix} 1.0\text{e}308 & 0 \\ 0 & 1.0\text{e-}150 \end{bmatrix}, \quad \mathrm{svd}(A) = \begin{bmatrix} 1.000000000000000\text{e}{+308} \\ 1.000036332667081\text{e-}150 \end{bmatrix},$$
$$B = \begin{bmatrix} 1.0\text{e}308 & 0 \\ 0 & 1.0\text{e-}155 \end{bmatrix}, \quad \mathrm{svd}(B) = \begin{bmatrix} 1.000000000000000\text{e}{+308} \\ 0 \end{bmatrix}.$$

The solution is a special implementation of Jacobi rotations in extremely ill-conditioned cases. The modified real Jacobi rotation from [18] can be adapted for complex matrices using the standard procedure [40], and the resulting Jacobi SVD algorithm computes the SVD with an accuracy described in Section 2.1 even for $\sigma_{\max}/\sigma_{\min} \approx 10^{616}$ with the 64-bit IEEE

ETNA
Kent State University
http://etna.math.kent.edu

double-precision arithmetic. In fact, even if the smallest singular values get denormalized, the algorithm will compute them with the relative accuracy degrading as they get deeper into the denormalized domain between zero and α. A sketch of the modified rotation is given in Algorithm 5. For more details, we refer to [18] and to the source code of the LAPACK routine `XGESVJ()`.[11]

Algorithm 5 $(s_i, s_j, \gamma_i, \gamma_j, \mathcal{J}) = \mathrm{JROT_X}(s_i, s_j, \gamma_i, \gamma_j, \gamma_{ij})$
(Jacobi rotation (s_i, s_j), $s_i, s_j \in \mathbb{C}^n$).

{**On input:** $\gamma_i = \|s_i\|_2$, $\gamma_j = \|s_j\|_2$, $\gamma_{ij} = s_i^* s_j/\gamma_i/\gamma_j$; assumed $\gamma_i \geq \gamma_j$ and $|\gamma_{ij}| > \widehat{\varepsilon}$}

$extreme = (\gamma_j/\gamma_i) > \alpha/\widehat{\varepsilon}$
{For a safe implementation of this line see `XGESVJ()` in `LAPACK`.}
if $\neg extreme$ **then**
  $\zeta = (\gamma_j/\gamma_i - \gamma_i/\gamma_j)/(2|\gamma_{ij}|)$; $\varphi = \gamma_{ij}/|\gamma_{ij}|$
  if $|\zeta| \leq 1/\sqrt{\widehat{\varepsilon}}$ **then**
    $\tau = \mathrm{sign}(\zeta)/(|\zeta| + \sqrt{1+\zeta^2})$; {τ is the tangent of the (smaller) Jacobi angle.}
  else
    $\tau = 0.5/\zeta$; {This is in particular important if ζ^2 overflows.}
  end if
  $cs = 1/\sqrt{1+\tau^2}$; $sn = \tau \cdot cs$; $\mathcal{J} = \begin{bmatrix} \varphi \cdot cs & \varphi \cdot sn \\ -sn & cs \end{bmatrix}$; $(s_i, s_j) = (s_i, s_j)\mathcal{J}$;
  $\gamma_j^{(c)} = \gamma_j$; $\gamma_j = \gamma_j \sqrt{1 + \tau \cdot (\gamma_i/\gamma_j) \cdot |\gamma_{ij}|}$; $\gamma_i = \gamma_i \sqrt{1 - \tau \cdot (\gamma_j^{(c)}/\gamma_i) \cdot |\gamma_{ij}|}$;
else
  $s_j = (\frac{s_j}{\gamma_j} - \gamma_{ij}(\frac{s_i}{\gamma_i}))\gamma_j$; $\gamma_j = \gamma_j \sqrt{1 - |\gamma_{ij}|^2}$.
  {(Re)scaled Gram-Schmidt orthogonalization.}
end if

EXAMPLE 5.4. As a demonstration of the numerical robustness of the algorithm and its implementation, we set[12] $n = 39$ and generate (implicitly) $\mathcal{H} = \mathcal{V}^T D \mathcal{V}$ such that $\sigma_{\max} \approx \omega$, $\sigma_{\min} \approx \alpha$, hence with a condition number close to $\omega/\alpha \approx 8 \cdot 10^{615}$. The condition number is $\sigma_1/\sigma_{39} \cong 0.28 \cdot 10^{615}$, and the results are presented in Figure 5.2. To our best knowledge, this is the highest reported condition number of a matrix whose all singular values and singular vectors are computed (to nearly 13 digits of accuracy!) using only the standard 16-digit IEEE double-precision arithmetic.

5.4.1. A comment on multiple-precision arithmetic. Suppose the input is given in, e.g., the standard IEEE double-precision with roundoff $\widehat{\varepsilon}$, and we need all singular values computed with small relative errors, roughly of the order of $\widehat{\varepsilon}$. If the singular values spread a large range of values, say $\kappa_2(A) \equiv \sigma_{\max}/\sigma_{\min} > 10^{500}$, and if multiple-precision arithmetic is available, then conventional SVD algorithms would require more than $\log_2(\kappa_2(A)/\widehat{\varepsilon})$ bits long mantissa to guarantee nearly working-precision accuracy of the computed singular values. Such multiple-precision arithmetic may exponentially increase the complexity of the computation, and, since it is software emulated, the run time increases by huge factors. Better algorithms should, ideally, be capable of delivering the required accuracy running in

[11] To our best knowledge, our Jacobi SVD is the only published algorithm capable of computing the singular values in the full range $[\alpha, \omega]$ of machine numbers.

[12] Here we take a moderate n because the reference values are computed in 640-digit arithmetic and experiments take a long time for large n.

ETNA
Kent State University
http://etna.math.kent.edu

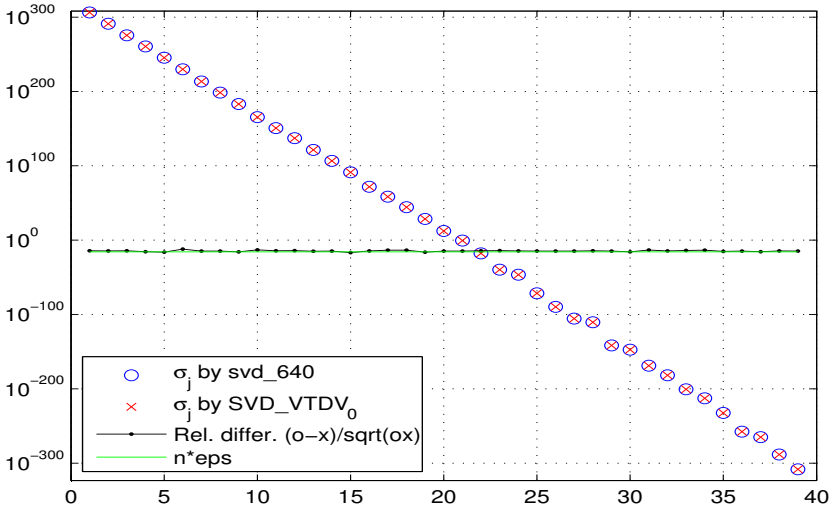

FIG. 5.2. *The singular values of the product* $\mathcal{H} = \mathcal{V}^T D\mathcal{V}$ *computed by Algorithm 3 in* 16*-digit arithmetic and the reference values computed in* 640*-digit arithmetic. The extreme singular values were* $\sigma_1 \approxeq 1.659563214356268e{+}306$, $\sigma_{39} \approxeq 5.752792768736278e{-}309$. *The maximal measured relative error was* $8.632997535220512e{-}013$.

the same working precision or at least to identify critical parts that require higher accuracy, not necessarily as high as dictated by $\kappa_2(A)$. This is a more economical way of using multi-precision arithmetic: instead of giving the number of digits that defines the arithmetic, the input is the required number of correct digits in the result. Then, the algorithm should determine what part of the code requires higher precision and what is the minimal higher precision that will suffice to deliver the result to the required accuracy. For example, in order to compute to ten decimal places all singular values of a matrix with the spectral condition number in the range of 10^{600}, it may be enough (using a robust algorithm) to run in the standard IEEE 16-digits arithmetic with the exception of a small non-iterative part of the computation that requires 32 or maybe 64 digits. This is obviously more efficient than having to run the whole algorithm with more than 600 digits. And, of course, the true challenge tackled in this paper is to have nearly working precision of accuracy without having to resort to higher-precision arithmetic.

6. Generalizations and limitations. Following the principles outlined in Section 2, the concept of RRD-based accurate SVD can be generalized to various product representations. For example, if $\mathcal{C}$ is an arbitrary Cauchy matrix, D_1, D_2 diagonal, and X, Y well-conditioned, then the SVD of $XD_1\mathcal{C}D_2Y^*$ uses a RRD $D_1\mathcal{C}D_2 = X_cD_cY_c^*$ and then applies Algorithm 2 to compute the SVD of $(XX_c)D_c(YY_c)^*$. This also applies if, instead of $\mathcal{C}$, we have $\mathcal{C}^T$, $\mathcal{C}^*$, or $\mathcal{C}^{-1}$. Also, if $\mathcal{V}$ is a Vandermonde matrix, then for the SVD of $XD_1\mathcal{V}Y^*$, we just apply the FFT to get $XD_1(\mathcal{V}\mathbb{F})(Y\mathbb{F})^* = X((D_1\tilde{D}_1)\mathcal{C}\tilde{D}_2)(Y\mathbb{F})^*$. The cases of $D_1\mathcal{C}^TD_3\mathcal{C}D_2$, $D_1\mathcal{C}^{-1}D_3\mathcal{C}^{-T}D_2$, $D_1\mathcal{C}^{-T}D_3\mathcal{C}^{-1}D_2$, and alike are also included (with D_1, D_2, D_3 arbitrary diagonal matrices).

For each of those cases, an analysis such as the one of Algorithm 3 in Section 5 must be performed to determine whether they are capable of delivering high accuracy. For the sake of brevity, we skip the details and provide in Section 6.1 below a nontrivial case study example to illustrate several interesting phenomena that could arise in a SVD computation of matrices given in product form. We use certain triangular Hankel matrices to study other

ETNA
Kent State University
http://etna.math.kent.edu

factored representations, different from (1.1), and to stress the importance of the conditions we identified in Section 5 as crucial for getting an accurate SVD.

6.1. A case study: triangular Hankel matrices in Cauchy-Vandermonde product form. Triangular Hankel matrices are the key objects in the Carathéodory-Fejér and the Takagi rational approximation theories. They are given by the Chebyshev (Fourier) coefficients $(h_i)_{i=1}^n \in \mathbb{C}^n$,

$$\mathcal{H}(h) = \begin{bmatrix} h_1 & h_2 & h_3 & \cdot & h_n \\ h_2 & h_3 & \cdot & h_n & 0 \\ h_3 & \cdot & \cdot & 0 & \cdot \\ \cdot & h_n & 0 & \cdot & 0 \\ h_n & 0 & \cdot & 0 & 0 \end{bmatrix}, \qquad \sum_{i=1}^{n} |h_i| > 0. \tag{6.1}$$

(Note that $|\det(\mathcal{H}(h))| = |h_n|^n$.) Takagi (re)discovered the symmetric SVD of complex symmetric matrices as a tool for generalizing the Carathéodory-Fejér results and showed how every singular vector of $\mathcal{H}(h)$ can be used to construct certain minimal meromorphic extensions of a polynomial on the unit disk [31, 45].

If only $h \in \mathbb{C}^n$ is given in (6.1), then $\mathcal{H}(h)$ can be represented via Cauchy and Vandermonde matrices using the relation (see [27, 39])

$$\begin{aligned} \mathcal{C}(x,y) &\equiv \left(\frac{1}{x_i - y_j}\right)_{i,j=1}^{n} = D_{xy}^{-1}\mathcal{V}(x)\mathcal{H}(h)\mathcal{V}(y)^T, \\ D_{xy} &= \operatorname{diag}(\prod_k (x_i - y_k))_{i=1}^n, \end{aligned} \tag{6.2}$$

where $x_i \neq y_j$ for all i, j, and $\chi_n(z) \equiv z^n + \sum_{i=1}^n h_i z^{i-1} = \prod_{i=1}^n (z - y_i)$, i.e., the y_i's are the zeros of the polynomial $\chi_n(z)$ defined by h. (The constraints $x_i \neq y_j$ can be removed by a proper reformulation using the matrix $D_{xy}\mathcal{C}(x,y)$; for simplicity let them be satisfied.)

Now take $\mathcal{V}(x)$ to be the Vandermonde matrix composed of the roots of unity, i.e., $x = (1, \omega, \ldots, \omega^{n-1})$ with $\omega = \mathbf{e}^{2\pi \mathrm{i}/n}$. Then, multiply the relation (6.2) by $1/\sqrt{n}$, set $\mathbb{F} = \mathcal{V}(x)/\sqrt{n}$, and pre-multiply by D_{xy} and then by $\mathbb{F}^*$ to obtain the identity $(1/\sqrt{n})\mathbb{F}^* D_{xy}\mathcal{C}(x,y) = \mathcal{H}(h)\mathcal{V}(y)^T$, where, using the formulas (4.1) for the DFT of a Vandermonde matrix,

$$\begin{aligned} \mathcal{V}(y)^T &= \mathbb{F}^{*T} D_c \mathcal{C}(-y, -x^{*T})^T D_r, \\ D_r &= \operatorname{diag}((1 - y_i^n)/\sqrt{n})_{i=1}^n, \quad D_c = \operatorname{diag}(1/\omega^{j-1})_{j=1}^n. \end{aligned}$$

This gives us two representations for $\mathcal{H}(h)$ using Cauchy and Vandermonde matrices,

$$\begin{aligned} \mathcal{H}(h) &= (1/\sqrt{n})\mathbb{F}^* D_{xy}\mathcal{C}(x,y)\mathcal{V}(y)^{-T} \\ &= (1/\sqrt{n})\mathbb{F}^* D_{xy}\mathcal{C}(x,y)D_r^{-1}\mathcal{C}(-y,-x^{*T})^{-T}D_c^{-1}\mathbb{F}^T. \end{aligned} \tag{6.3}$$

From now on, we consider only the right-hand sides of the equality signs in (6.3) and consider the problem of computing their SVD's as functions of x and y. Implicitly, this is the SVD of the matrix $\mathcal{H}(h)$ that is given by the zeros y_i of the polynomial $\chi_n(z)$, where, by our simplifying assumption, none of the y_i is a root of unity. (For a given h, the y_i's can be computed, e.g., by a polynomial root finder in extended precision and then rounding to working precision, thus implicitly changing h to $h + \delta h$; see Remark 4.3.)

ETNA
Kent State University
http://etna.math.kent.edu

6.1.1. Algorithms based on the Cauchy–Vandermonde product in (6.3). Consider the first product representation in (6.3). Compute the RRD $D_{xy}\mathcal{C}(x,y) = X\Delta Y^*$ and then, using the accurate Vandermonde SVD algorithm of Demmel [13], the SVD $\mathcal{V}(y) = U\Sigma W^*$ to obtain

$$\mathcal{H}(h) = \frac{1}{\sqrt{n}}(\mathbb{F}^*X)\Delta(Y^*U^{*T})\Sigma^{-1}W^T = \frac{1}{\sqrt{n}}\mathbb{F}(X\Delta)Y_1\Sigma^{-1}W^T, \quad Y_1 = Y^*U^{*T}.$$

Since $\mathbb{F}$ and W are unitary and the factor $1/\sqrt{n}$ only scales the singular values, the problem reduces to computing the SVD of $A = X\Delta Y_1\Sigma^{-1}$. Note that here the SVD of $\mathcal{V}(y)$ can be ordered so that $\operatorname{diag}(\Sigma^{-1})$ is non-increasing and that any reordering permutation of the singular values also permutes the columns of U and thus the columns of Y_1. We will try two different approaches to compute the SVD of A. In both cases, the SVD of the product $\mathcal{H}(h)$ is obtained by scaling the singular values and a straightforward assembling of the singular vectors. We will not go into the details of the error analysis; instead we will briefly discuss the key issues, illustrate them using numerical examples that show both, successes and failures, and leave the full analysis as a challenge for future work.

(CV1). (i) Use the LDU decomposition with full pivoting to compute the RRD $X_BD_BY_B^*$ of $B = \Delta Y_1\Sigma^{-1}$. (ii) Then, apply Algorithm 2 and compute the SVD of the product $(XX_B)D_BY_B^*$.

What can we expect from this procedure? Looking back at the analysis of the pivoted LDU of $A = D_4ZD_4$ in Section 5.1.2 and Section 5.2.3, where the matrix $Z = L^TL$ had certain cross-product structure reinforced by the structure of L and where the scaling by D_4 was symmetric with decreasing diagonal entries (in absolute values), here we have very little to start with. Here, for a successful RRD of $\Delta Y_1\Sigma^{-1}$, we hope (cf. [15, Section 4]) for both, well-conditioned leading minors of Y_A and

$$\tau(\Delta,\Sigma^{-1}) = \max\{\max_{i<j}|\Delta_{jj}/\Delta_{ii}|, \max_{i<j}|(\Sigma^{-1})_{jj}/(\Sigma^{-1})_{ii}|\}$$

not much bigger than one, and that the complete pivoting will automatically and implicitly reorder B into such a structure (possibly implicitly defining better diagonal scaling matrices). Whether these conditions will be satisfied, at least nearly, is uncertain as well as is then the accuracy of the computed SVD. In the examples in Section 6.1.3, it frequently happens (depending on the distribution of y) that only one of these conditions is fulfilled while the other one is violated causing disappointingly bad results.

(CV2). The following procedure involves more computation and is not practical, but it is included for the sake of an experiment. (i) Let $(X\Delta)P = QR$ be a rank-revealing QR factorization (e.g., with Businger-Golub column pivoting). (ii) Then the problem reduces to the SVD of $M \equiv RP^TY_1\Sigma^{-1} \equiv S(TP^TY_1)\Sigma^{-1}$, where[13] $R = ST$ with a diagonal S and a well-conditioned T. The SVD of M is obtained by first computing its RRD $M = X_MD_MY_M^*$ and then invoking Algorithm 2.

6.1.2. Algorithms based on the Cauchy-Cauchy product in (6.3). To exploit the second relation algorithmically, we use [27] to write $\mathcal{C}(-y,-x^{*T})^{-1} = -D_1\mathcal{C}(-y,-x^{*T})^TD_2$, where

$$D_1 = \operatorname{diag}\left(\frac{\prod_{j=1}^n(y_j-\overline{x_i})}{\prod_{\substack{j=1\\j\neq i}}^n(\overline{x_j}-\overline{x_i})}\right)_{i=1}^n, \quad D_2 = \operatorname{diag}\left(\frac{\prod_{j=1}^n(\overline{x_j}-y_i)}{\prod_{\substack{j=1\\j\neq i}}^n(y_j-y_i)}\right)_{i=1}^n,$$

[13] Here, by $R = ST$, we only indicate that R has a certain structure. The matrices S and T are not computed in the algorithm.

ETNA
Kent State University
http://etna.math.kent.edu

and then $\mathcal{H}(h) = -(1/\sqrt{n})\mathbb{F}D_{xy}\mathcal{C}(x,y)D_r^{-1}D_2\mathcal{C}(-y,-x^{*T})D_1D_c^{-1}\mathbb{F}^T$. As in Section 6.1.1, the problem can be reduced to the SVD of $D_{xy}\mathcal{C}(x,y)D_r^{-1}D_2\mathcal{C}(-y,-x^{*T})D_1$. Now we can proceed as follows:

(CC1). (i) Compute the RRD's (based on the LDU [13])

$$M \equiv D_{xy}\mathcal{C}(x,y)\sqrt{D_r^{-1}D_2} = X_M D_M Y_M^*,$$
$$N \equiv \sqrt{D_r^{-1}D_2}\mathcal{C}(-y,-x^{*T})D_1 = X_N D_N Y_N^*.$$

(ii) Then, compute a RRD $X_Z D_Z Y_Z^*$ of $D_M Y_M^* X_N D_N$. (iii) Compute the SVD of $(X_M X_Z)D_Z(Y_N Y_Z)^*$.

(CC2). Follow the same path as in **(CC1)** but with

$$M = D_{xy}\mathcal{C}(x,y)D_r^{-1}, \quad N = D_2\mathcal{C}(-y,-x^{*T})D_1.$$

Note that further variations of the above procedures are possible. The reader is invited to formulate and test them in numerical experiments as described below.

6.1.3. Numerical examples. Only error and perturbation analysis can give an estimate of the numerical accuracy of a proposed algorithm. And, as we indicated in the discussion following the description of **(CV1)**, which applies to all algorithms described in this section, such an analysis does not seem feasible. It is, however, clear that in each particular case, the distribution of the vector y will play an important role. In the following two examples, we illustrate this dependence and, at the same time, we show different phenomena that leave many open questions for future work. The interested reader will repeat the experiments and perform a diagnostic research to explain some of the results. We leave this as a research problem.

EXAMPLE 6.1. In the first numerical test, we generated with $n = 50$ a complex vector of the y_i's as $y = 20y_{re} + 20\mathrm{i}y_{im}$, where the entries of y_{re}, y_{im} are chosen randomly from the normal distribution $\mathcal{N}(0,1)$. The results of all four algorithms are displayed in the first plot in Figure 6.1. Only **(CC2)** has failed to compute all singular values to high relative accuracy, and the problem can be traced to the known critical part—the RRD of $D_M Y_M^* X_N D_N$. If only that part of the algorithm is computed in extended precision (to compensate ill-conditioning of the leading minors of $Y_M^* X_N$) and the RRD is rounded back to working precision, then the final result is as accurate as in the remaining three algorithms. If the entries of y_{re}, y_{im} are chosen randomly from the uniform $\mathcal{U}(0,1)$ distribution, then the outcome is less satisfactory. As shown in the second plot in Figure 6.1, not even the dominant singular values are computed with an relative error below 0.1. As much as this is disappointing, it is instructive because it shows that all singular values, including the largest, may be lost. We refer the reader to [15, Section 4] for a more detailed discussion on this nontrivial issue.

EXAMPLE 6.2. In this example, we set $n = 136$, and the y_i's are taken to be the Fekete points for the square $[-1,1] \times [-1,1]$ and then the Lebesgue points for the unit disk both generated using [8]. The results are given in Figure 6.2. In the first plot (Fekete points), interestingly, the smallest singular values are better approximated than the largest ones. The corresponding Hankel matrix is in this case not severely ill-conditioned (as compared to our previous examples) with $\kappa_2(\mathcal{H}) \approxeq 8.6308\text{e+}31$. The second plot (Lebesgue points) shows another interesting phenomenon. The underlying Hankel matrix (the product in (6.3)) is well-conditioned, $\kappa_2(\mathcal{H}) \approxeq 7.1172\text{e+}01$, and yet none of the four methods could capture five digits of accuracy in any singular value. At the same time, the results computed by **(CV1)** and **(CV2)** coincide to fourteen decimal digits. (Recall the discussion in the first paragraph of Section 4.3.1.)

ETNA
Kent State University
http://etna.math.kent.edu

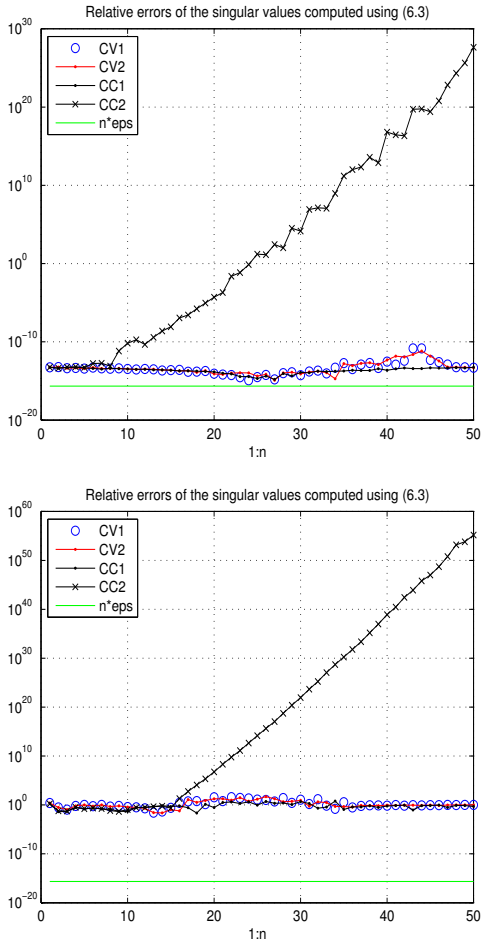

FIG. 6.1. *Measured relative errors of the algorithms described in Section 6.1.1 and Section 6.1.2 with input data as explained in Example 6.1. The dimension is set to* $n = 50$*, and the reference values are computed in* 330*-digits arithmetic. In the first plot (top),* $\kappa_2(\mathcal{H}) \approxeq 1.3251e{+}155$ *(*$\sigma_1 \approx 1.0754e{+}67$*,* $\sigma_{50} \approx 8.1159e{-}89$*). In the second plot (bottom),* $\kappa_2(\mathcal{H}) \approxeq 4.9210e{+}144$ *(*$\sigma_1 \approx 5.8855e{+}58$*,* $\sigma_{50} \approx 1.1960e{-}86$*).*

7. Concluding remarks. The key for developing an accurate SVD algorithm for certain ill-conditioned but structured matrices is to consider matrix representations in a wider sense than a mere two-dimensional array. In some cases, the most important step is not to form the matrix at all. For, computing and storing the entries of an ill-conditioned matrix will inevitably introduce rounding errors which, no matter how tiny, may irreparably destroy information on the smallest singular values or eigenvalues. Instead, such as, e.g., in the case of Cauchy and Vandermonde matrices, a natural parametrization is used to compute—in a forward stable way—their rank revealing decomposition (RRD) as a product of factors that are either (i) well-conditioned, generally dense, and non-structured, or (ii) diagonal and arbitrarily ill-conditioned. Then, a Jacobi-type SVD procedure is deployed to compute an accurate SVD of such a product independent of the ill-conditioning carried by the diagonal factors.

In this paper, we have explored possibilities to compute to high accuracy all singular values and vectors of a Hankel matrix that is given in a factored form $\mathcal{H} = \mathcal{V}(x)^T \mathrm{diag}(d) \mathcal{V}(x)$, where $\mathcal{V}(x)$, $\mathrm{diag}(d)$ are the Vandermonde and the diagonal matrix defined by the vectors x

ETNA
Kent State University
http://etna.math.kent.edu

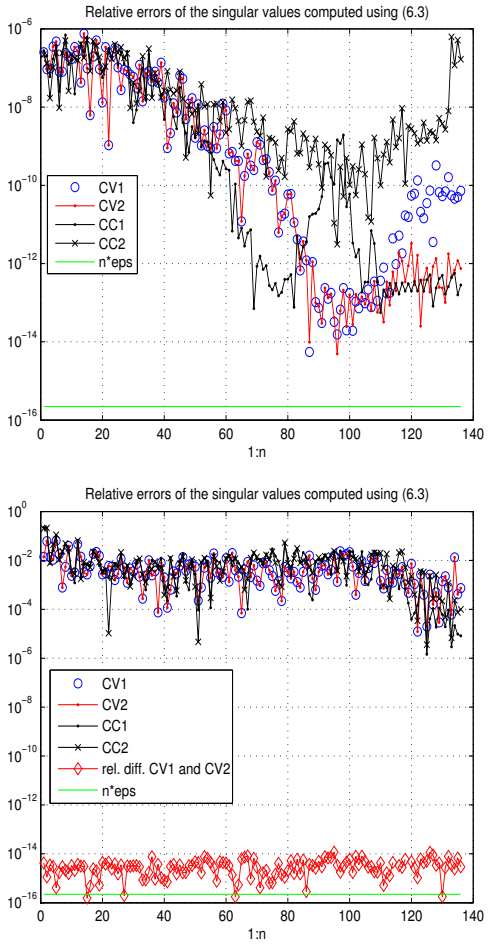

FIG. 6.2. *Measured relative errors of the algorithms described in Section 6.1.1 and Section 6.1.2 with input data as explained in Example 6.2. The reference values are computed in* 330*-digits arithmetic. In the top plot (Fekete points),* $\kappa_2(\mathcal{H}) \approxeq 8.6308e{+}31$ *(*$\sigma_1 \approx 2.0360e{+}07$*,* $\sigma_{136} \approx 2.3590e{-}25$*). Note that the smallest singular values are computed to high accuracy and more accurately than the largest ones. In the second plot (Lebesgue points),* $\kappa_2(\mathcal{H}) \approxeq 7.1172e{+}01$ *(*$\sigma_1 \approx 6.4283e{+}00$*,* $\sigma_{136} \approx 9.0321e{-}02$*). Note that* ***(CV1)*** *and* ***(CV2)*** *agree to nearly fourteen digits but only up to five digits are actually correct. Let us finally mention that for, e.g.,* $n = 55$*, all methods were very accurate and that for* $n = 91$ *all of them computed nearly eight accurate digits in every singular value.*

and d, respectively. The proposed algorithms follow the ideas introduced in [13, 15]. We have shown how to accurately reduce an RRD with several factors to a *well-conditioned* $\times$ *diagonal* matrix product by using certain scaling-invariance properties of the QR factorization and the LU decomposition. This is the key for the final step of computing the SVD by the Jacobi SVD algorithm, whose accuracy is independent of the diagonal scaling. Also, we have provided detailed error and perturbation analyses with computable relevant condition numbers which allow a reliable estimate of the accuracy of the computed SVD. The numerical experiments match the theoretical predictions.

Altogether, we believe that, in addition to the new core numerical algebra algorithms, we have provided a new tool for computational rational approximations and an instructive case study with all the details of the development of a robust numerical algorithm and its

ETNA
Kent State University
http://etna.math.kent.edu

software implementation. The limitations of the proposed approach, illustrated in Section 6, pose several interesting research challenges for the future work.

Acknowledgements. I would like to thank the referees for detailed reports, constructive comments, and suggestions. Thanks as well to Dr. Christopher Beattie and Dr. Serkan Gugercin for their hospitality during my stay at the Virginia Tech and to Pavel Holoborodko for discussions related to the Advanpix toolbox [1].

REFERENCES

[1] ADVANPIX, *Multiprecision Computing Toolbox for MATLAB, v3.6.3.4942*, Advanpix LLC., Yokohama, Japan, April 2014.
http://www.advanpix.com/
[2] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. D. CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNY, S. OSTROUCHOV, AND D. SORENSEN, *LAPACK Users' Guide,*, 2nd ed., SIAM, Philadelphia, 1992.
[3] L. AUTONNE, *Sur les matrices hypohermitiennes et sur les matrices unitaires*, Ann. Univ. Lyon, Nouvelle Sér. I, 38 (1915), pp. 1–77.
[4] J. L. BARLOW, *More accurate bidiagonal reduction for computing the singular value decomposition*, SIAM J. Matrix Anal. Appl, 23 (1998), pp. 761–798.
[5] A. BARRLUND, *Perturbation bounds for the LDL^H and LU decompositions*, BIT, 31 (1991), pp. 358–363.
[6] B. BECKERMANN, *The condition number of real Vandermonde, Krylov and positive definite Hankel matrices*, Numer. Math., 85 (2000), pp. 553–577.
[7] D. L. BOLEY, F. T. LUK, AND D. VANDEVOORDE, *Vandermonde factorization of a Hankel matrix*, in Scientific Computing: Proceedings of the 6th Workshop held at Hong Kong Baptist University, G. H. Golub, S. H. Lui, F. T. Luk, and R. J. Plemmons, eds., Springer, Singapore, 1997, pp. 27–39.
[8] M. BRIANI, A. SOMMARIVA, AND M. VIANELLO, *Computing Fekete and Lebesgue points: Simplex, square, disk*, J. Comput. Appl. Math., 236 (2012), pp. 2477–2486.
[9] P. A. BUSINGER AND G. H. GOLUB, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.
[10] A. J. COX AND N. J. HIGHAM, *Stability of Householder QR factorization for weighted least squares problems*, in Numerical Analysis 1997, Proceedings of the 17th Dundee Biennial Conference, D. F. Griffiths, D. J. Higham, and G. A. Watson, eds., vol. 380 of Pitman Research Notes in Mathematics, Longman, Harlow, 1998, pp. 57–73.
[11] J. DANCIGER, *A min-max theorem for complex symmetric matrices*, Linear Algebra Appl., 412 (2006), pp. 22–29.
[12] J. DEMMEL, *On condition numbers and the distance to the nearest ill–posed problem*, Numer. Math., 51 (1987), pp. 251–289.
[13] ———, *Accurate singular value decompositions of structured matrices*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 562–580.
[14] J. DEMMEL, I. DUMITRIU, AND O. HOLTZ, *Toward accurate polynomial evaluation in rounded arithmetic*, in Foundations of Computational Mathematics, L. M. Pardo, A. Pinkus, E. Süli, and M. J. Todd, eds., London Mathematical Society Lecture Note Series, 331, Cambridge University Press, Cambridge, 2006, pp. 36–105.
[15] J. DEMMEL, M. GU, S. EISENSTAT, I. SLAPNIČAR, K. VESELIĆ, AND Z. DRMAČ, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.
[16] J. DEMMEL AND P. KOEV, *Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials*, Linear Algebra Appl., 417 (2006), pp. 382–396.
[17] J. DEMMEL AND K. VESELIĆ, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.
[18] Z. DRMAČ, *Implementation of Jacobi rotations for accurate singular value computation in floating point arithmetic*, SIAM J. Sci. Comput., 18 (1997), pp. 1200–1222.
[19] ———, *Accurate computation of the product induced singular value decomposition with applications*, SIAM J. Numer. Anal., 35 (1998), pp. 1969–1994.
[20] ———, *New accurate algorithms for singular value decomposition of matrix triplets*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1026–1050.
[21] ———, *On principal angles between subspaces of Euclidean space*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 173–194.
[22] Z. DRMAČ AND Z. BUJANOVIĆ, *On the failure of rank revealing QR factorization software—a case study*, ACM Trans. Math. Software, 35 (2008), Art. 12 (28 pages).

ETNA
Kent State University
http://etna.math.kent.edu

[23] Z. DRMAČ AND K. VESELIĆ, *New fast and accurate Jacobi SVD algorithm I.*, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1322–1342.
[24] ———, *New fast and accurate Jacobi SVD algorithm II.*, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1343–1362.
[25] S. EISENSTAT AND I. IPSEN, *Relative perturbation techniques for singular value problems*, SIAM J. Numer. Anal., 32 (1995), pp. 1972–1988.
[26] S. FELDMANN AND G. HEINIG, *Vandermonde factorization and canonical representations of block Hankel matrices*, Linear Algebra Appl., 241/243 (1996), pp. 247–278.
[27] T. FINCK, G. HEINIG, AND K. ROST, *An inversion formula and fast algorithms for Cauchy-Vandermonde matrices*, Linear Algebra Appl., 183 (1993), pp. 179–191.
[28] P. A. FUHRMANN, *Remarks on the inversion of Hankel matrices*, Linear Algebra Appl., 81 (1986), pp. 89–104.
[29] P. GONNET, R. PACHÓN, AND L. N. TREFETHEN, *Robust rational interpolation and least-squares*, Electron. Trans. Numer. Anal., 38 (2011), pp. 146–167.
http://etna.mcs.kent.edu/vol.38.2011/pp146-167.dir
[30] W. GRAGG AND L. REICHEL, *On singular values of Hankel operators of finite rank*, Linear Algebra Appl., 121 (1989), pp. 53–70.
[31] M. H. GUTKNECHT, *Rational Carathéodory-Fejér approximation on a disk, a circle, and an interval*, J. Approx. Theory, 41 (1984), pp. 257–278.
[32] M. H. GUTKNECHT AND L. N. TREFETHEN, *Real polynomial Chebyshev approximation by the Carathéodory–Fejér method*, SIAM J. Numer. Anal., 19 (1982), pp. 358–371.
[33] T. S. HAUT AND G. BEYLKIN, *Fast and accurate con-eigenvalue algorithm for optimal rational approximations.*, SIAM J. Matrix Anal. Appl.,, 33 (2012), pp. 1101–1125.
[34] D. J. HIGHAM, *Condition numbers and their condition numbers*, Linear Algebra Appl., 214 (1995), pp. 193–213.
[35] N. J. HIGHAM, *Factorizing complex symmetric matrices with positive definite real and imaginary parts*, Math. Comp, 67 (1998), pp. 1591–1599.
[36] ———, *QR factorization with complete pivoting and accurate computation of the SVD*, Linear Algebra Appl., 309 (2000), pp. 153–174.
[37] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.
[38] R.-C. LI, *Relative perturbation theory: II. Eigenspace and singular subspace variations*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 471–492.
[39] V. Y. PAN, M. A. TABANJEH, Z. Q. CHEN, E. I. LANDOWNE, AND A. SADIKOU, *New transformations of Cauchy matrices and Trummer's problem*, Comput. Math. Appl., 35 (1998), pp. 1–5.
[40] H. PARK AND V. HARI, *A real algorithm for the Hermitian eigenvalue decomposition*, BIT, 33 (1993), pp. 158–171.
[41] M. J. D. POWELL AND J. K. REID, *On applying Householder transformations to linear least squares problems*, in Information Processing 68, Proc. IFIP, Edinburgh, Vol. 1, A. J. H. Morrell, ed., 1968, North Holland, Amsterdam, 1969, pp. 122–126.
[42] I. SCHUR, *Ein Satz ueber quadratische Formen mit komplexen Koeffizienten*, Amer. J. Math., 67 (1945), pp. 472–480.
[43] G. W. STEWART, *Perturbation theory for the singular value decomposition*, Tech. Report UMIACS-TR-90-124, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, 1990.
[44] V. STRASSEN, *Gaussian elimination is not optimal*, Numer. Math., 13 (1969), pp. 354–356.
[45] T. TAKAGI, *On an algebraic problem related to an analytic theorem of Carathéodory and Fejér and on an allied theorem of Landau*, Japan. J. Math., 1 (1924), pp. 83–93.
[46] T. TAO AND V. H. VU, *The condition number of a randomly perturbed matrix*, in Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing, STOC '07, U. Feige, ed., ACM, New York, 2007, pp. 248–255.
[47] L. N. TREFETHEN AND R. S. SCHREIBER, *Average-case stability of Gaussian elimination*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 335–360.
[48] E. V. TYRTYSHNIKOV, *How bad are Hankel matrices?*, Numer. Math., 67 (1994), pp. 261–269.
[49] A. VAN DER SLUIS, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.
[50] K. VESELIĆ AND V. HARI, *A note on a one–sided Jacobi algorithm*, Numer. Math., 56 (1989), pp. 627–633.
[51] D. VISWANATH AND L. N. TREFETHEN, *Condition numbers of random triangular matrices*, SIAM J. Matrix. Anal. Appl., 19 (1998), pp. 564–581.
[52] W. XU AND S. QIAO, *A fast symmetric SVD algorithm for square Hankel matrices*, Linear Algebra Appl., 428 (2008), pp. 550–563.
[53] K. YE AND L.-H. LIM, *Every matrix is a product of Toeplitz matrices*, Preprint on arXiv, (2013).
http://arxiv.org/abs/1307.5132