COMPUTING SYMMETRIC RANK-REVEALING DECOMPOSITIONS VIA TRIANGULAR FACTORIZATION*

PER CHRISTIAN HANSEN† AND PLAMEN Y. YALAMOV‡

Abstract. We present a family of algorithms for computing symmetric rank-revealing VSV decompositions, based on triangular factorization of the matrix. The VSV decomposition consists of a middle symmetric matrix that reveals the numerical rank in having three blocks with small norm, plus an orthogonal matrix whose columns span approximations to the numerical range and null space. We show that for semi-definite matrices the VSV decomposition should be computed via the ULV decomposition, while for indefinite matrices it must be computed via a URV-like decomposition that involves hyperbolic rotations.

Key words. rank-revealing decompositions, matrix approximation, symmetric matrices

AMS subject classifications. 65F30, 65F35

1. Introduction.

Rank-revealing decompositions of general dense matrices are widely used in signal processing and other applications where accurate and reliable computation of the numerical rank, as well as the numerical range and null space, are required. The singular value decomposition (SVD) is certainly a decomposition that reveals the numerical rank, but what we have in mind here are the RRQR and UTV (i.e., URV and ULV) decompositions which can be computed and, in particular, updated more efficiently than the SVD. See, e.g., [7, §§2.7.5–2.7.7] and [19] for details and references to theory, algorithms, and applications.

The key to the efficiency of RRQR and UTV algorithms is that they consist of an initial triangular factorization which can be tailored to the particular matrix, followed by a rank-revealing post-processing step. If the matrix is $m \times n$ with $m \geq n$ and with numerical rank k, then the initial triangular factorization requires $\mathcal{O}(mn^2)$ flops, while the rank-revealing step only requires $\mathcal{O}((n-k)n^2)$ flops if $k \approx n$, and $\mathcal{O}(kn^2)$ flops if $k \ll n$. The updating can always be done in $\mathcal{O}(n^2)$ flops, when implemented properly. We refer to the original papers [8], [9], [15], [17], [18], [22], [30], [31] for details about the algorithms.

For structured matrices (e.g., Hankel and Toeplitz matrices), the initial triangular factorization in the RRQR and UTV algorithms has the same complexity as the rank-revealing step, namely, $\mathcal{O}(mn)$ flops; see [7, §8.4.2] for signal processing aspects. However, accurate principal singular values and vectors can also be computed by means of Lanczos methods in the same complexity, $\mathcal{O}(mn)$ flops [12]. Hence the advantage of a rank-revealing decomposition depends on the matrix structure and the numerical rank of the matrix.

Rank-revealing decompositions of general *sparse* matrices are also in use, e.g., in optimization and geometric design [26]. For sparse matrices, the initial pivoted triangular factorization can exploit the sparsity of A. However, the UTV post-processors may produce a severe amount of fill, while the fill in the RRQR post-processor is

*P. Y. Yalamov was supported by a fellowship from the Danish Rectors' Conference and by Grants MM-707/97 and I-702/97 from the National Scientific Research Fund of the Bulgarian Ministry of Education and Science.

†Per Christian Hansen, Department of Mathematical Modelling, Technical University of Denmark, Building 321, DK-2800 Lyngby, Denmark (pch@imm.dtu.dk).

‡Center of Applied Mathematics and Informatics, University of Rousse, 7017 Rousse, Bulgaria (yalamov@ami.ru.acad.bg).

restricted to lie in the columns that are permuted to the right of the triangular factor [7, Thm. 6.7.1]. An alternative sparse URL decomposition $A = U\,R\,L$, where U is orthogonal and R and L are upper and lower triangular, respectively, was proposed in [25]. This decomposition can be computed with less fill, at the expense of working with only one orthogonal matrix.

Numerically rank-deficient *symmetric* matrices also arise in many applications, notably in signal processing and in optimization algorithms (such as those based on interior point and continuation methods). In both areas, fast computation and efficient updating are key issues, and sparsity is also an issue in some optimization problems. Utilization of symmetry leads to faster algorithms, compared to algorithms for nonsymmetric matrices. In addition, symmetric rank-revealing decompositions enable us to compute symmetric rank-deficient matrix approximations (obtained by neglecting blocks in the rank-revealing decomposition with small norm). This is important, e.g., in rank-reduction algorithms in signal processing where one wants to compute rank-deficient symmetric semidefinite matrices.

In spite of this, very little work has been done on symmetric rank-revealing decompositions. Luk and Qiao [23] introduced the term *VSV decomposition* and proposed an algorithm for symmetric indefinite Toeplitz matrices, while Baker and DeGroat [2] presented an algorithm for symmetric semi-definite matrices.

The purpose of this paper is to expand on the ideas in [2] and [23] and present a broader survey of possible rank-revealing VSV decompositions and algorithms, including the underlying theory. Our emphasis is on algorithms which, in addition to revealing the numerical rank, provide accurate estimates of the numerical range and null space. We build our algorithms on existing methods for computing rank-revealing decompositions of triangular matrices, based on orthogonal transformations. Our symmetric decompositions and algorithms inherit the properties of these underlying algorithms which are well understood today.

We emphasize that the goal of this paper is not to present detailed implementations of our VSV algorithms, but rather to set the stage for such implementations. The papers [4] and [27] clearly demonstrate that careful implementations of efficient and robust mathematical software for numerically rank-deficient problems requires a major amount of research which is outside the scope of the present paper.

Our paper is organized as follows. After briefly surveying general rank-revealing decompositions in §2, we define and analyze the rank-revealing VSV decomposition of a symmetric matrix in §3. Numerical algorithms for computing VSV decompositions of symmetric semi-definite and indefinite matrices are presented in §4, and we conclude with some numerical examples in §5.

2. General Rank-Revealing Decompositions. In this paper we restrict our attention to real square $n \times n$ matrices. The singular value decomposition (SVD) of a square matrix is given by

$$A = U\,\Sigma\,V^T = \sum_{i=1}^{n} u_i\,\sigma_i\,v_i^T\,, \tag{2.1}$$

where u_i and v_i are the columns of the orthogonal matrices U and V, and $\Sigma = \mathrm{diag}(\sigma_i)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. Then $\|A\|_2 = \sigma_1$, $\|A\|_F^2 = \sum_{i=1}^{n} \sigma_i^2$, and $\mathrm{cond}(A) = \sigma_1/\sigma_n$. The numerical rank k of A, with respect to the threshold τ, is the number of singular values greater than or equal to τ, i.e., $\sigma_k \geq \tau > \sigma_{k+1}$ [19, §3.1].

The RRQR, URV, and ULV decompositions are given by

$$A = Q\,T\,\Pi^T = U_R\,R\,V_R^T = U_L\,L\,V_L^T\,.$$

Here, Q, U_R, U_L, V_R, and V_L are orthogonal matrices, Π is a permutation matrix, T and R are upper triangular matrices, and L is a lower triangular matrix. Moreover, if we partition the triangular matrices as

$$T = \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix}, \qquad R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}, \qquad L = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix},$$

then the numerical rank k of A is revealed in the triangular matrices in the sense that T_{11}, R_{11}, and L_{11} are $k \times k$ and

$$\mathrm{cond}(T_{11}) \simeq \sigma_1/\sigma_k, \qquad \|T_{22}\|_F^2 \simeq \sigma_{k+1}^2 + \cdots + \sigma_n^2$$

$$\mathrm{cond}(R_{11}) \simeq \sigma_1/\sigma_k, \qquad \|R_{12}\|_F^2 + \|R_{22}\|_F^2 \simeq \sigma_{k+1}^2 + \cdots + \sigma_n^2$$

$$\mathrm{cond}(L_{11}) \simeq \sigma_1/\sigma_k, \qquad \|L_{21}\|_F^2 + \|L_{22}\|_F^2 \simeq \sigma_{k+1}^2 + \cdots + \sigma_n^2.$$

The first k columns of the left matrices Q, U_R, and U_L span approximations to the numerical range of A, defined as $\mathrm{span}\{u_1, \ldots, u_k\}$, and the last $n-k$ columns of the right matrices V_R and V_L span approximations to the numerical null-space of A, defined as $\mathrm{span}\{v_{k+1}, \ldots, u_n\}$. See, e.g., [19, §3.1] for details.

Precise definitions of RRQR decompositions and algorithms are given by Chandrasekaran and Ipsen [10], Gu and Eisenstat [18] and Hong and Pan [22], and associated large-scale implementations are available in Fortran [4]. Definitions of UTV decompositions and algorithms are given by Stewart [30], [31]. Matlab software for both RRQR and UTV decompositions is available in the UTV Tools package [16].

3. Symmetric Rank-Revealing Decompositions. For a symmetric $n \times n$ matrix A, we need rank-revealing decompositions that inherit the symmetry of the original matrix. In particular this is true for the eigenvalue decomposition (EVD)

$$A = V\,\Lambda\,V^T = \sum_{i=1}^{n} v_i\,\lambda_i\,v_i^T\;, \tag{3.1}$$

where v_i are the right singular vectors, while $\sigma_i = |\lambda_i|$ and $u_i = \mathrm{sign}(\lambda_i)\,v_i$ for $i = 1, \ldots, n$.

Corresponding to the UTV decompositions, Luk and Qiao [23] defined the following VSV decomposition

$$A = V_S\,S\,V_S^T, \tag{3.2}$$

where V_S is an orthogonal matrix, and S is a symmetric matrix with partitioning

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{12}^T & S_{22} \end{pmatrix}, \tag{3.3}$$

in which S_{11} is $k \times k$. We say that the VSV decomposition is rank-revealing if

$$\mathrm{cond}(S_{11}) \simeq \sigma_1/\sigma_k, \qquad \|S_{12}\|_F^2 + \|S_{22}\|_F^2 \simeq \sigma_{k+1}^2 + \cdots + \sigma_n^2.$$

This definition is very similar to the definition used by Luk and Qiao, except that they use $\|\mathrm{triu}(S_{22})\|_F^2$ instead of $\|S_{22}\|_F^2$, where "triu" denotes the upper triangular part. Our choice is motivated by the fact that $\|S_{22}\|_F^2 \to \sigma_{k+1}^2 + \cdots + \sigma_n^2$ as $\|S_{12}\|_F \to 0$.

Given the VSV decomposition in (3.2), the first k columns of V_S and the last $n-k$ columns of V_S provide approximate basis vectors for the numerical range and null space, respectively. Moreover, given the ill-conditioned problem $A\,x = b$, we can compute a stabilized "truncated VSV solution" x_k by neglecting the three blocks in S with small norm, i.e., $x_k = V_{S,k} S_{11}^{-1} V_{S,k}^T b$ where $V_{S,k}$ consists of the first k columns of V_S. We return to the computation of x_k in §4.4.

Instead of working directly with the matrix S, it is more convenient to work with a symmetric decomposition of S and, in particular, of S_{11}. The form of this decomposition depends on both the matrix A (semi-definite or indefinite) and the rank-revealing algorithm. Hence, we postpone a discussion of the particular form of S to the presentation of the algorithms. Instead, we summarize the approximation properties of the VSV decomposition.

THEOREM 3.1. *Let the EVD and the VSV decompositions of A be given by (3.1) and (3.2), respectively, and partition the matrix S as in (3.3). Then the singular values $\bar\sigma_i$ of* $\mathrm{diag}(S_{11}\,,\,S_{22})$ *are related to those of A as*

$$|\bar\sigma_i - \sigma_i| \le \|S_{12}\|_2, \qquad i = 1,\ldots,k. \tag{3.4}$$

Moreover, the angle Θ between the subspaces spanned by the first k columns of V and V_S is bounded as

$$\frac{\|S_{12}\|_2}{\sigma_1 + \sigma_{k+1}} \le \sin\Theta \le \frac{\|S_{12}\|_2}{\sigma_k - \|S_{22}\|_2}. \tag{3.5}$$

Proof. The bound (3.4) follows from the standard perturbation bound for singular values:

$$|\bar\sigma_i - \sigma_i| \le \left\| \begin{pmatrix} 0 & S_{12} \\ S_{12}^T & 0 \end{pmatrix} \right\|_2 = \|S_{12}\|_2 \ ,$$

where we use that the singular values of the symmetric "perturbation matrix" appear in pairs. To prove the upper bound in (3.5), we partition $V = (V_k\,,\,V_0)$ and $V_S = (V_{S,k}\,,\,V_{S,0})$ such that V_k and $V_{S,k}$ have k columns. Moreover, we write $\Lambda = \mathrm{diag}(\Lambda_k\,,\,\Lambda_0)$ where Λ_k is $k \times k$. If we insert these partitionings as well as (3.1) and (3.2) into the product $A\,V_{S,0}$ then we obtain

$$\left(V_k \Lambda_k\, V_k^T + V_0\, \Lambda_0\, V_0^T\right) V_{S,0} = V_{S,k}\, S_{12} + V_{S,0}\, S_{22}.$$

Multiplying from the left with V_k^T we get

$$\Lambda_k\, V_k^T\, V_{S,0} = V_k^T V_{S,k}\, S_{12} + V_k^T V_{S,0}\, S_{22}$$

from which we obtain

$$V_k^T V_{S,0} = \Lambda_k^{-1}\left(V_k^T V_{S,k}\, S_{12} + V_k^T V_{S,0}\, S_{22}\right).$$

Taking norms in this expression and inserting $\sin\Theta = \|V_k^T V_{S,0}\|_2$ and $\|\Lambda_k^{-1}\|_2 = \sigma_k^{-1}$, we get

$$\sin\Theta \le \sigma_k^{-1}\|S_{12}\|_2 + \sigma_k^{-1}\|S_{22}\|_2 \sin\Theta$$

which immediately leads to the upper bound in (3.5). To prove the lower bound, we use that

$$S_{12} = V_{S,k}^T A\, V_{S,0} = V_{S,k}^T V_k \Lambda_k V_k^T V_{S,0} + V_{S,k}^T V_0 \Lambda_0 V_0^T V_{S,0}.$$

Taking norms and using $\sin\Theta = \|V_k^T V_{S,0}\|_2 = \|V_{S,k}^T V_0\|_2$, $\|\Lambda_k\|_2 = \sigma_1$ and $\|\Lambda_0\|_2 = \sigma_{k+1}$, we obtain the left bound in (3.5). ∎

We conclude that if there is a well-defined gap between σ_k and σ_{k+1}, and if the norm $\|S_{12}\|_2$ of the off-diagonal block is sufficiently small, then the numerical rank k is indeed revealed in S, and the first k columns of V_S span an approximation to the singular subspace $\mathrm{span}\{v_1, \ldots, v_k\}$. The following theorem shows that a well-defined gap is also important for the perturbation bounds.

THEOREM 3.2. *Let $\widetilde{A} = A + \Delta A = \widetilde{V}_S\, \widetilde{S}\, \widetilde{V}_S^T$, and let Φ denote the angle between the subspaces spanned by the first k columns of V_S and $\widetilde{V}_S$; then*

$$\sin\Phi \le \frac{4\tau + \|\Delta A\|_2}{\sigma_k - \sigma_{k+1} - 4\tau - \|\Delta A\|_2}, \tag{3.6}$$

where $\tau = \max\{\|S_{12}\|_2, \|\widetilde{S}_{12}\|_2\}$.

Proof. The bound follows from Corollary 3.2 in [13]. ∎

4. Algorithms for Symmetric Rank-Revealing Decompositions. Similar to general rank-revealing algorithms, the symmetric algorithms consist of an initial triangular factorization and a rank-revealing post-processing step. The purpose of the latter step is to ensure that the largest k singular values are revealed in the leading submatrix S_{11} and that the corresponding singular subspace is approximated by the span of the first k columns of V_S.

For a semi-definite matrix A, our initial factorization is the symmetrically pivoted Cholesky factorization

$$P^T A\, P = C^T C, \tag{4.1}$$

where P is the permutation matrix, and C is the upper triangular (or trapezoidal) Cholesky factor. The numerical properties of this algorithm are discussed by Higham in [21]. If A is a symmetric semi-definite Toeplitz matrix, then there is good evidence (although no strict proof) that the Cholesky factor can be computed efficiently and reliably without the need for pivoting by means of the standard Schur algorithm [29].

If A is indefinite, then our initial factorization is the symmetrically pivoted LDL^T factorization

$$P^T A\, P = L\, D\, L^T, \tag{4.2}$$

where P is the permutation matrix, L is a unit lower triangular matrix, and D is a block diagonal matrix with 1×1 and 2×2 blocks on the diagonal. The state-of-the-art in LDL^T algorithms is described in [1], where it is pointed out that special care must be taken in the implementation to avoid large entries in L when A is ill conditioned. Alternatively, one could use the $G\Omega G^T$ factorization described in [28]. If A is a symmetric indefinite Toeplitz matrix, then the currently most reliable approach to computing the LDL^T factorization seems to be via orthogonal transformation to a Cauchy matrix [20].

The reason why we need the post-processing step is that the initial factorization may not reveal the numerical rank of A — there is no guarantee that small eigenvalues

TABLE 4.1
The four post-processing rank-revealing steps for a symmetric semi-definite matrix.

Post-proc.	Decomposition	Symmetric matrix
URV	$C = U_R\, R\, V_R^T$	$S = R^T R = \begin{pmatrix} R_{11}^T R_{11} & R_{11}^T R_{12} \\ R_{12}^T R_{11} & R_{12}^T R_{12} + R_{22}^T R_{22} \end{pmatrix}$
RRQR	$C = Q\, T\, \Pi^T$	$S = T^T T = \begin{pmatrix} T_{11}^T T_{11} & T_{11}^T T_{12} \\ T_{12}^T T_{11} & T_{12}^T T_{12} + T_{22}^T T_{22} \end{pmatrix}$
ULV	$ECE = U_L\, L\, V_L^T$	$S = L^T L = \begin{pmatrix} L_{11}^T L_{11} + L_{21}^T L_{21} & L_{21}^T L_{22} \\ L_{22}^T L_{21} & L_{22}^T L_{22} \end{pmatrix}$
RRQR	$(ECE)^T = Q\, T\, \Pi^T$	$S = T\, T^T = \begin{pmatrix} T_{11} T_{11}^T + T_{12}\, T_{12}^T & T_{12} T_{22}^T \\ T_{22} T_{12}^T & T_{22} T_{22}^T \end{pmatrix}$

of A manifest themselves in small diagonal elements of C or in small eigenvalues of D. In particular, since $\|A^{-1}\|_2 = \sigma_n^{-1} \leq \|L^{-1}\|_2^2\, \|D^{-1}\|_2 = \sigma_n(L)^{-2}\, \sigma_n(D)^{-1}$ and $\sigma_n \leq \sigma_n(D)\, \|L\|_2^2$, we obtain

$$\sigma_n(L)^2 \leq \frac{\sigma_n}{\sigma_n(D)} \leq \|L\|_2^2$$

showing that a small σ_n may not be revealed in D when L is ill conditioned.

4.1. Algorithms for Semi-Definite Matrices. For symmetric semi-definite matrices there is a simple relationship between the SVDs of A and C.

THEOREM 4.1. *The right singular vectors of $P^T A\, P$ are also the right singular vectors of C, and*

$$\sigma_i(C)^2 = \sigma_i, \qquad i = 1, \ldots, n. \tag{4.3}$$

Proof. The result follows directly from inserting the SVD of C into $A = C^T C$.

Hence, once we have computed the initial pivoted Cholesky factorization (4.1), we can proceed by computing a rank-revealing decomposition of C, and this can be done in several ways. Let E denotes the exchange matrix consisting of the columns of the identity matrix in reverse order, and write $P^T A\, P$ as

$$P^T A\, P = C^T C = E\, (ECE)^T\, (ECE)\, E.$$

Then we can compute a URV or RRQR decomposition of C, a ULV decomposition of ECE, or an RRQR decomposition of $(ECE)^T$, as shown in the left part of Table 4.1. The approach using the URV decomposition of C was suggested in [2]. Table 4.1 also shows the particular forms of the resulting symmetric matrix S, as derived from the following relations:

$$\begin{aligned} P^T A\, P &= V_R\, R^T R\, V_R^T && \text{(URV post-processor)} \\ &= \Pi\, T^T T\, \Pi^T && \text{(RRQR post-processor)} \\ &= (E\, V_L)\, L^T L\, (E\, V_L)^T && \text{(ULV post-processor)} \\ &= (E\, Q)\, T\, T^T (E\, Q)^T && \text{(RRQR post-processor.)} \end{aligned}$$

Three of these four approaches lead to a symmetric matrix S that reveals the numerical rank of A by having both an off-diagonal block and a bottom right block with small norm. This is, however, not the case for the approach based on RRQR decomposition of the Cholesky factor C. Instead, since T_{11} is well conditioned, this algorithm provides a symmetric permutation $P\Pi$ that is guaranteed to produce a well-conditioned leading $k \times k$ submatrix in $(P\Pi)^T A\,(P\Pi)$.

The remaining three algorithms yield approximate bases for the range and null spaces of A, due to Theorem 3.1. It is well known that among the rank-revealing decompositions, the ULV decomposition can be expected to provide the most accurate bases for the right singular subspaces, in the form of the columns of V_L; see, e.g., [31] and [14]). Therefore, the algorithm that computes the ULV decomposition of ECE is to be preferred. We remark that the matrix U_L in the ULV decomposition need not be computed.

In terms of the blocks S_{12} and S_{22}, the ULV-based algorithm is the only algorithm that guarantees small norms of both the off-diagonal block $S_{12} = L_{21}^T L_{22}$ and the bottom right block $S_{22} = L_{22}^T L_{22}$, because the norms of both L_{12} and L_{22} are guaranteed to be small. From Theorem 4.1 and the definition of the ULV decomposition we have $\|L_{21}\|_2 \simeq \|L_{22}\|_2 \simeq \sigma_{k+1}^{1/2}$ and therefore $\|S_{12}\|_2 \simeq \|S_{22}\|_2 \simeq \sigma_{k+1}$.

For a *sparse* matrix the situation is different, because the UTV post-processors may produce severe fill, while the RRQR post-processor produces only fill in the $n-k$ rightmost columns of T. For example, if A is the upper bidiagonal matrix

$$A = \begin{pmatrix} 10^{-5} B_{n-k} & e_k e_1^T \\ 0 & B_k \end{pmatrix},$$

in which B_p is an upper bidiagonal $p \times p$ matrix of all ones, and e_p is the pth column of the identity matrix, then URV with threshold $\tau = 10^{-4}$ produces a full $k \times k$ upper triangular R_{11}, while RRQR with the same threshold produces a $k \times k$ upper bidiagonal T_{11}. Hence, for sparsity reasons, the UTV approaches may not be suited for computing the VSV decomposition, depending on the sparsity pattern of A.

An alternative is to use the algorithm based on RRQR decomposition of the transposed and permuted Cholesky factor $(ECE)^T = E\,C^T E$, and we note that the permutation matrix Π is not needed. In terms of the matrix S, only the bottom right submatrix of S is guaranteed to have a norm of the order σ_{k+1}, because of the relations $\|S_{12}\|_2 = \|T_{12} T_{22}^T\|_2 \simeq \sigma_1^{1/2} \sigma_{k+1}^{1/2}$ and $\|S_{22}\|_2 = \|T_{22} T_{22}^T\|_2 \simeq \sigma_{k+1}$.

In practice the situation is often better, because the RRQR-algorithm—when applied to the matrix $E\,C^T E$—tends to produce an off-diagonal block T_{12} whose norm is smaller than what is guaranteed (namely, of the order $\sigma_1^{1/2}$). The reason is that the initial Cholesky factor C often has a trailing $(n-k) \times (n-k)$ triangular block C_{22} whose norm is close to $\sigma_{k+1}^{1/2}$, which may produce a norm $\|S_{12}\|_2$ close to σ_{k-1}. From the partitionings

$$C = \begin{pmatrix} C_{11} & C_{12} \\ 0 & C_{22} \end{pmatrix}, \qquad E\,C^T E = \begin{pmatrix} E_{n-k} C_{22}^T E_{n-k} & E_{n-k} C_{12}^T E_k \\ 0 & E_k C_{11}^T E_k \end{pmatrix}$$

and the fact that the RRQR post-processor leaves column norms unchanged and is likely to permute the leading $n-k$ columns of $E\,C^T E$ to the back, we see that the norm of the resulting off-diagonal block T_{12} in the RRQR decomposition is likely to be bounded by $\|C_{22}\|_2$. Our numerical examples in §5 illustrate this.

However, we stress that in the RRQR approach we can only guarantee that $\|S_{12}\|_2$ is of the order $\sigma_1^{1/2} \sigma_{k+1}^{1/2}$, and this point is illustrated by the matrix $A = K^T K$, where K is the "infamous" Kahan matrix [7, p. 105] that is left unchanged by QR factorization with ordinary column pivoting, yet its numerical rank is $k = n - 1$. Cholesky factorization with symmetric pivoting computes the Cholesky factor $C = K$, and when we apply RRQR to $E\, C^T E$ we obtain an upper triangular matrix T in which only the (n, n)-element is small, while $\|T_{12}\|_2 = 1 \simeq \|T\|_2$ and $\|S_{12}\|_2 = \|T_{12}\, T_{22}^T\|_2 \simeq \|T_{22}\|_2 \simeq \sigma_n^{1/2}$.

4.2. Algorithms for Indefinite Matrices. All known rank-revealing post-processors maintain the triangular form of the matrix in consideration, but when we apply them to the matrix L in the LDL^T factorization we destroy the block diagonal form of D. We avoid this difficulty by inserting an additional *interim stage* between the initial LDL^T factorization and the rank-revealing post-processor, in which the middle block-diagonal matrix D is replaced by a signature matrix Ω, i.e., a diagonal matrix whose diagonal elements are ± 1. At the same time, L is replaced by the product of an orthogonal matrix and a triangular matrix. The interim processor computes the factorization

$$P^T A\, P = \overline{W}\, C^T \Omega\, C\, \overline{W}^T \tag{4.4}$$

where $\overline{W}$ is orthogonal and C is upper triangular, and it takes the following generic form.

Interim Processor for Symmetric Indefinite Matrices

1. Compute the eigenvalue decomposition $D = W \Lambda\, W^T$.
2. Write Λ as $\Lambda = |\Lambda|^{1/2}\, \Omega\, |\Lambda|^{1/2}$.
3. Compute an orthogonal $\overline{W}$ such that $C^T = \overline{W}^T L\, W\, |\Lambda|^{1/2}$ is lower triangular.

The interim processor is simple to implement and requires at most $\mathcal{O}(n^2)$ overhead, because W and $\overline{W}$ are block diagonal matrices with the same block structure as D. For each 1×1 block d_{ii} in D the corresponding 1×1 blocks in W, $|\Lambda|^{1/2}$, and $\overline{W}$ are equal to 1, $|d_{ii}|^{1/2}$, and 1, respectively. For each 2×2 block in D we compute the eigenvalue decomposition

$$\begin{pmatrix} d_{ii} & d_{i,i+1} \\ d_{i,i+1} & d_{i+1,i+1} \end{pmatrix} = W_{ii} \begin{pmatrix} \lambda_i & 0 \\ 0 & \lambda_{i+1} \end{pmatrix} W_{ii}^T \ ;$$

then the corresponding 2×2 block in W is W_{ii}, and the associated 2×2 block in $\overline{W}$ is a Givens rotation chosen such that C stays triangular. If A is sparse, then some fill may be introduced in C by the interim processor, but since the Givens transformations are applied to nonoverlapping 2×2 blocks, fill introduced in the treatment of a particular block does not spread during the processing of the other blocks. The same type of interim processor can also be applied to the $G\Omega G^T$ factorization in [28].

We shall now explore the possibilities for using post-processors similar to the ones for semi-definite matrices, but modified such that they yield a rank-revealing decomposition in which either the leftmost or rightmost matrix M is orthogonal with respect to the signature matrix Ω, i.e., we require $M^T \Omega\, M = \hat{\Omega}$, where $\hat{\Omega}$ is also a signature matrix. Note that in general we cannot guarantee that $\hat{\Omega} = \Omega$.

One possibility would be to compute an RRQR-like decomposition $C = Q\, T\, \Pi^T$ with $Q^T \Omega\, Q = \hat{\Omega}$, but none of the blocks in the resulting $S = R^T \hat{\Omega}\, R$ are guaranteed to have small norm, neither are we guaranteed to obtain a symmetrically permuted A with a well-conditioned leading $k \times k$ submatrix. It is an open question how to compute such a symmetric permutation for an indefinite matrix.

We now turn to algorithms that produce submatrices in S with small norm. The following theorem shows that there is hope such algorithms will exist.

THEOREM 4.2. *If $\sigma_n(C)$ denotes the smallest singular value of C in the interim factorization (4.4), then*

$$\sigma_n(C) \leq \sigma_n^{1/2} \tag{4.5}$$

Proof. We have $\sigma_n^{-1} = \|(C^T \Omega C)^{-1}\|_2 \leq \|C^{-T}\|_2 \|\Omega\|_2 \|C^{-1}\|_2 = \|C^{-1}\|_2^2 = \sigma_n(C)^{-2}$, from which the result follows. $\square$

This theorem shows that a small singular value of A is guaranteed to be revealed in the triangular matrix C. Unfortunately, there is no guarantee that $\sigma_n(C)$ does not underestimate $\sigma_n^{1/2}$ dramatically, neither does it ensure that the size of σ_n is revealed in S. Hence, for indefinite matrices we cannot rely solely on the matrix C, and the following theorem (which expands on results in [23]) shows how to proceed instead.

THEOREM 4.3. *Let w_n be the eigenvector of $C^T \Omega\, C$ corresponding to the eigenvalue λ_n that is smallest in absolute value, and let $\tilde{w}_n$ be an approximation to w_n. Moreover, choose the orthogonal matrix V_S such that $V_S^T \tilde{w}_n = e_n$, the last column of the identity matrix, and partition the matrix*

$$V_S^T C^T \Omega\, C\, V_S = S = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix}$$

such that S_{11} is $(n-1) \times (n-1)$. Then

$$\|s_{12}\|_2 \leq (\sigma_1 + \sigma_n)\, \tan\phi \tag{4.6}$$

and

$$|s_{22} - \lambda_n| \leq (\sigma_1 + \sigma_n)\, \tan\phi, \tag{4.7}$$

where ϕ is the angle between w_n and $\tilde{w}_n$.

Proof. Consider first the quantity

$$V_S^T A\, \tilde{w}_n = S\, V_S^T \tilde{w}_n = S\, e_n = \begin{pmatrix} s_{12} \\ s_{22} \end{pmatrix}.$$

Next, write $\tilde{w}_n = w_n + u$ to obtain

$$\begin{aligned} V_S^T A\, \tilde{w}_n &= V_S^T A\, (w_n + u) = \lambda_n\, V_S^T w_n + V_S^T A\, u \\ &= \lambda_n\, V_S^T (\tilde{w}_n - u) + V_S^T A\, u = \lambda_n\, e_n - \lambda_n\, V_S^T u + V_S^T A\, u. \end{aligned}$$

Combining these two results we obtain

$$\begin{pmatrix} s_{12} \\ s_{22} - \lambda_n \end{pmatrix} = V_S^T (A - \lambda\, I)\, u$$

and taking norms we get

$$\|s_{12}\|_2^2 + (s_{22} - \lambda_2)^2 = \|((A - \lambda\, I)\, u\|_2^2 \leq \|A - \lambda\, I\|_2^2\, \|u\|_2^2.$$

Both $\|s_{12}\|_2^2$ and $|s_{22} - \lambda_2|$ are lower bounds for the left-hand side, while $\|u\|_2$ is bounded above by $\tan\phi$. Combining this with the bound $\|A - \lambda\, I\|_2 \leq \sigma_1 + \sigma_n$ we obtain the two bounds in the theorem. $\square$

The above theorem shows that in order for σ_n to reveal itself in S, we must compute an approximate null vector of $C^T \Omega C$, apply Givens rotations to this vector to transform it into e_n, and accumulate these rotations from the right into C. At the same time, we should apply hyperbolic rotations from the left in order to keep C triangular. Theorem 4.3 ensures that $\|s_{12}\|_2$ is small and that s_{22} approximates λ_n. We note that hyperbolic transformations can be numerically unstable, and in our implementations we use stabilized hyperbolic rotations [7, §3.3.4].

Once this step has been performed, we deflate the problem and apply the same technique to the $(n-1)\times(n-1)$ submatrix $S_{11} = \hat{C}_{11}^T \hat{\Omega}_{11} \hat{C}_{11}$, where $\hat{C}_{11}$ and $\hat{\Omega}_{11}$ are the leading submatrices of the updated factors. This is precisely the algorithm from [23]. When the process stops (because all the small singular values of A are revealed) we have computed the URV-like decomposition $C = U_R R V_R^T$ such that $U_R^T \Omega U_R = \widehat{\Omega}$, and the middle rank-revealing matrix is given by

$$S = R^T \widehat{\Omega} R = \begin{pmatrix} R_{11}^T \widehat{\Omega}_1 R_{11} & R_{11}^T \widehat{\Omega}_1 R_{12} \\ R_{12}^T \widehat{\Omega}_1 R_{11} & R_{12}^T \widehat{\Omega}_1 R_{12} + R_{22}^T \widehat{\Omega}_2 R_{22} \end{pmatrix} \tag{4.8}$$

where $\widehat{\Omega} = \mathrm{diag}(\widehat{\Omega}_1\,,\,\widehat{\Omega}_2)$ and $\widehat{\Omega}_1$ is $k \times k$.

The condition estimator used in the URV-like post-processor must be modified, compared to the standard URV algorithm, because we must now estimate the smallest singular value of the matrix $C^T \Omega C$. In our implementation we use one step of inverse iteration applied to $C^T \Omega C$, with starting vector from the condition estimator of the ordinary URV algorithm applied to C.

The ULV algorithm cannot be modified analogously, the reason being that the left matrix U_L must transform the approximate left singular vector into the form e_n. Hence, U_L is an orthogonal matrix, but it is not orthogonal with respect to Ω, and this rules out the use of a ULV-like approach for symmetric indefinite matrices.

Finally, we consider the use of the RRQR decomposition of $(ECE)^T$, which can be used without modification because the product $\Pi^T E \Omega E \Pi$ remains a signature matrix. This approach is more appealing for sparse problems because the RRQR decomposition preserves sparsity better than the above URV-like approach. There is, however, no guarantee that this approach will work because it relies solely on revealing small singular values of C.

We illustrate this with a small 5×5 numerical example from [1] where A is given by $A = L\,D\,L^T$ with

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & -\frac{20}{13} & -\frac{8}{17} & 1 & 0 \\ 1 & \frac{6\cdot 10^6}{13} & -\frac{1}{17} & 0 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{10^{-19}}{3} & \frac{6\cdot 10^{-7}}{7} & 0 & 0 \\ 0 & \frac{6\cdot 10^{-7}}{7} & -\frac{3\cdot 10^{-6}}{13} & 0 & 0 \\ 0 & 0 & 0 & -\frac{4\cdot 10^{-5}}{17} & \frac{2}{7} \\ 0 & 0 & 0 & \frac{2}{7} & \frac{1}{300} \end{pmatrix}$$

and $\mathrm{cond}(L) = 3.01 \cdot 10^{11}$. The singular values of A are

$$\sigma_1 = 5.13, \quad \sigma_2 = 0.270, \quad \sigma_3 = 0.142, \quad \sigma_4 = 2.66 \cdot 10^{-7}, \quad \sigma_5 = 1.14 \cdot 10^{-8}$$

such that A has full rank with respect to the threshold $\tau = 10^{-10}$. The corresponding matrix C has singular values

$$\sigma_1(C) = 104, \quad \sigma_2(C) = 2.02, \quad \sigma_3(C) = 0.459,$$

$$\sigma_4(C) = 3.10 \cdot 10^{-4}, \quad \sigma_5(C) = 8.17 \cdot 10^{-7}.$$

Thus, if we use the threshold $\tau^{1/2} = 10^{-5}$ in the RRQR decomposition of C we wrongly conclude that A is numerically rank deficient. The algorithm based on the URV-like approach, on the other hand, reveals the correct numerical rank.

The above example shows that the numerical rank of C may be smaller than that of A. The following example shows that the opposite may also be the case:

$$R = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \widehat{\Omega} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad R\widehat{\Omega}R^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

in which $\mathrm{rank}(R) = 2 > \mathrm{rank}(A) = 1$. Althoug this is an extreme case, it signals that difficulties may also arise in the near-rank-deficient case.

To summarize, for symmetric indefinite matrices only the approach using the URV-like post-processor is guaranteed to reveal the numerical rank of A.

4.3. Updating the VSV Decomposition. One of the advantages of the rank-revealing VSV decomposition over the EVD and SVD is that it can be updated efficiently when A is modified by a rank-one change $v\,v^T$. From the relation

$$A^{\mathrm{up}} = A + v\,v^T = V_S\left(S + (V_S^T v)(V_S^T v)^T\right) V_S^T$$

we see that the updating of A amounts to updating the rank-revealing matrix S by the rank-one matrix $w\,w^T$ with $w = V_S^T v$, i.e., $S^{\mathrm{up}} = S + w\,w^T$.

Consider first the semi-definite case, and let M denote one of the triangular matrices R, L, or T^T from the algorithms in Table 4.1. Then

$$S^{\mathrm{up}} = M^T M + w\,w^T = \begin{pmatrix} M \\ w^T \end{pmatrix}^T \begin{pmatrix} M \\ w^T \end{pmatrix}$$

and we see that the VSV updating is identical to standard updating of a triangular RRQR or UTV factor, which can be done stably and efficiently by means of Givens transformation as described in [5], [30] and [31].

Next we consider the indefinite case (4.8), where the updating takes the form

$$S^{\mathrm{up}} = R^T\widehat{\Omega}R + w\,w^T = \begin{pmatrix} R \\ w^T \end{pmatrix}^T \begin{pmatrix} \widehat{\Omega} & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} R \\ w^T \end{pmatrix},$$

showing that the VSV updating now involves hyperbolic rotations. Hence, the updating is computationally similar to UTV downdating, whose stable implementation is discussed in [3] and [24]. Downdating the VSV decomposition will, in both cases, also involve hyperbolic rotations and a signature matrix.

4.4. Computation of Truncated VSV Solutions. Here we briefly consider the computation of the truncated VSV solution which we define as

$$V_{S,k}\, S_{11}^{-1} V_{S,k}^T b, \tag{4.9}$$

where $V_{S,k}$ consists of the first k columns of V_S. For the URV-based decomposition $S_{11} = R_{11}^T R_{11}$ and $S_{11}^{-1} = \left(R_{11}^{-1}\right)^T R_{11}^{-1}$. For the ULV-based decomposition we have $S_{11} = L_{11}^T L_{11} + L_{21}^T L_{21}$, but we can safely neglect the term $L_{21}^T L_{21}$ whose norm is at most of the same order as the neglected blocks S_{12} and S_{22}, namely, σ_{k+1}. Finally, for the RRQR-based decomposition we can use the following theorem.

THEOREM 4.4. *If $S = T\,T^T$ and $\widehat{T}$ is the triangular QR factor of $(T_{11}\,,\,T_{12})^T$ then*

$$S_{11}^{T} = \widehat{T}^{-1}\big(\widehat{T}^{-1}\big)^T. \tag{4.10}$$

Alternatively, if the columns of the matrix

$$W = \begin{pmatrix} W_1 \\ W_2 \end{pmatrix}, \qquad W_1 \in \mathbb{R}^{(n-k)\times(n-k)}$$

form an orthonormal basis for the null space of $(T_{11}\,,\,T_{12})$ then

$$S_{11}^{-1} = \big(T_{11}^{-1}\big)^T \left(I - W_1\,W_1^T\right) T_{11}^{-1}. \tag{4.11}$$

Proof. If $(T_{11}\,,\,T_{12})^T = \widehat{Q}\,\widehat{T}$ is a QR factorization then $S_{11} = \widehat{T}^T\widehat{T}$ and $S_{11}^{-1} = \widehat{T}^{-1}\big(\widehat{T}^{-1}\big)^T$ whici is (4.10). The same relation leads to $S_{11}^{-1} = \widehat{T}^{-1}\widehat{Q}^T\widehat{Q}\,\big(\widehat{T}^{-1}\big)^T = \big((T_{11}\,,\,T_{12})^\dagger\big)^T\,(T_{11}\,,\,T_{12})^\dagger$, where † denotes the pseudoinverse. In [6] used that

$$(T_{11}\,,\,T_{12})^\dagger = (I - W\,W^T)\,T^{-1}\begin{pmatrix} I_1 \\ 0 \end{pmatrix}$$

which, combined with the relation $(I - W\,W^T)^2 = I - W\,W^T$, immediately leads to (4.11). ☐

The first relation (4.10) in Theorem 4.4 can be used when $k \ll n$, while the second relation (4.11) is more useful when $k \approx n$. Note that W can be computed by orthonormalization of the columns of the matrix

$$Z = \begin{pmatrix} R_{11}^{-1}R_{12} \\ -I \end{pmatrix}.$$

This approach is particularly useful for sparse matrices because we only introduce fill when working with the "skinny" $n \times (n-k)$ matrix Z.

5. Numerical Examples. The purpose of this section is to illustrate the theory derived in the previous sections by means of some test problems. Although robustness, efficiency and flop counts are important practical issues, they are also tightly connected to the particular implementation of the rank-revealing post-processor, and not the subject of this paper.

All our experiments were done in Matlab, and we use the implementations of the ULV, URV, and RRQR algorithms from the UTV TOOLS package [16]. The condition estimation in all three implementations is the Cline-Conn-Van Loan (CCVL) estimator [11]. The modified URV algorithm used for symmetric indefinite matrices is based on the URV algorithm from [16], augmented with stabilized hyperbolic rotations when needed, and with a condition estimator consisting of the CCVL algorithm followed by one step of inverse iteration applied to the matrix $C^T\Omega\,C$.

Numerical results for all the rank-revealing algorithms are shown in Table 5.1, where we present mean and maximum values of the norms of various submatrices associated with the VSV decompositions. In particular, X_{off} denotes either R_{12}, L_{21}, or T_{12}, and X_{22} denotes either R_{22}, L_{22}, or T_{22}. The results are computed on the

TABLE 5.1
Numerical results for the rank-revealing VSV algorithms.

Post-processor		$\|\|X_{\mathrm{off}}\|\|_2$	$\|\|X_{22}\|\|_2$	$\|\|S_{12}\|\|_2$	$\|\|S_{22}\|\|_2$
URV	mean	$2.2 \cdot 10^{-4}$	$3.2 \cdot 10^{-4}$	$6.5 \cdot 10^{-6}$	$2.5 \cdot 10^{-7}$
(semi-def.)	max	$3.0 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$8.1 \cdot 10^{-5}$	$9.4 \cdot 10^{-6}$
ULV	mean	$2.7 \cdot 10^{-7}$	$3.2 \cdot 10^{-4}$	$8.6 \cdot 10^{-11}$	$1.0 \cdot 10^{-7}$
(semi-def.)	max	$4.7 \cdot 10^{-7}$	$3.2 \cdot 10^{-4}$	$1.5 \cdot 10^{-10}$	$1.0 \cdot 10^{-7}$
RRQR	mean	$1.5 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$4.8 \cdot 10^{-7}$	$1.0 \cdot 10^{-7}$
(semi-def.)	max	$2.9 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$9.2 \cdot 10^{-7}$	$1.0 \cdot 10^{-7}$
URV-like	mean	$1.4 \cdot 10^{-4}$	$3.1 \cdot 10^{-4}$	$4.6 \cdot 10^{-4}$	$2.3 \cdot 10^{-7}$
(indef.)	max	$2.1 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$9.4 \cdot 10^{-3}$	$4.3 \cdot 10^{-6}$

TABLE 5.2
Numerical results when the CCVL estimate is improved by one inverse iteration step.

Matrix type		$\|\|R_{12}\|\|_2$	$\|\|R_{22}\|\|_2$	$\|\|S_{12}\|\|_2$	$\|\|S_{22}\|\|_2$
Semi-	mean	$1.8 \cdot 10^{-7}$	$3.1 \cdot 10^{-4}$	$2.3 \cdot 10^{-9}$	$1.0 \cdot 10^{-7}$
definite	max	$5.1 \cdot 10^{-6}$	$3.2 \cdot 10^{-4}$	$6.1 \cdot 10^{-8}$	$1.0 \cdot 10^{-7}$
Indefinite	mean	$6.9 \cdot 10^{-10}$	$3.1 \cdot 10^{-4}$	$4.5 \cdot 10^{-9}$	$1.0 \cdot 10^{-7}$
	max	$2.3 \cdot 10^{-8}$	$3.2 \cdot 10^{-4}$	$1.5 \cdot 10^{-7}$	$1.0 \cdot 10^{-7}$

basis of randomly generated test matrices of size 64, 128, and 256 (100 matrices of each size), each with $n-4$ eigenvalues geometrically distributed between 1 and 10^{-4}, and the remaining four eigenvalues given by 10^{-7}, 10^{-8}, 10^{-9}, and 10^{-10}, such that the numerical rank with respect to the threshold $\tau = 10^{-5}$ is $k = n-4$.

The test matrices were produced by generating random orthogonal matrices and multiplying them to diagonal matrices with the desired eigenvalues. For the indefinite matrices the signs of the eigenvalues were chosen to alternate.

Table 5.1 illustrates the superiority of the ULV-based algorithm for semi-definite matrices, for which the norm $||S_{12}||_2$ of the off-diagonal block in S is always much smaller than the norm $||S_{22}||_2$ of the bottom right submatrix. This is due to the fact that the ULV algorithm produces a lower triangular matrix L whose off-diagonal block L_{21} has a very small norm (and we emphasize that the size of this norm depends on the condition estimator). The second best algorithm for semi-definite matrices is the one based on the RRQR algorithm, for which $||S_{12}||_2$ and $||S_{22}||_2$ are of the same size. Note that it is the latter algorithm which we recommend for sparse matrices. The URV-based algorithm for semi-definite matrices produces results that are consistently less satisfactory than the other two algorithms. All these results are consistent with our theory.

For the indefinite matrices, only the URV-like algorithm can be used, and the results in Table 5.1 show that this algorithm also behaves as expected from the theory. In order to judge the backward stability of this algorithm, which uses hyperbolic rotations, we also computed the backward error $||A - V_S S V_S^T||_2$ for all three hundred test problems. The largest residual norm was $1.9 \cdot 10^{-11}$, and the average is $1.5 \cdot 10^{-12}$. We conclude that we loose a few digits of accuracy due to the use of the hyperbolic rotations.

It is well known that the norm of the off-diagonal block in the triangular URV factor depends on the quality of the condition estimator — the better the singular vector estimate, the smaller the norm. Hence, it is interesting to see how much the norms of the off-diagonal blocks in R and S decrease if we improve the singular vector estimates by means of one step of inverse iteration (at the expense of additional $2(n-k)n^2$ flops). In the semi-definite case we now apply an inverse iteration step to the CCVL estimate, and in the indefinite case we use two steps of inverse iteration applied to $C^T \Omega C$ instead of one. The results are shown in Table 5.2 for 100 matrices of size $n = 256$. As expected, the norms of the off-diagonal blocks are now smaller, at the expense of more work. The average backward errors $\|A - V_S S V_S^T\|_2$ did not change in this experiment.

6. Conclusion. We have defined and analyzed a class of rank-revealing VSV decompositions for symmetric matrices, and proposed algorithms for computing these decomposition. For semi-definite matrices, the ULV-based algorithm is the method of choice for dense matrices, while the RRQR-based algorithm is better suited for sparse matrices because it preserves sparsity better. For indefinite matrices, only the URV-based algorithm is guaranteed to work.

REFERENCES

[1] C. Ashcraft, R. G. Grimes, and J. G. Lewis, *Accurate Symmetric Indefinite Linear Equation Solvers*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 513–561.

[2] E. S. Baker and R. D. De Groat, *A correlation-based subspace tracking algorithm*, IEEE Trans. Signal Proc., 46 (1998), pp. 3112–3116.

[3] J. L. Barlow, P. A. Yoon, and H. Zha, *An algorithm and a stability theory for downdating the ULV decomposition*, BIT, 36 (1996), pp. 15–40.

[4] C. H. Bischof and G. Quintana-Ortí, *Algorithm 781: Codes for computing rank-revealing QR factorizations of dense matrices*, ACM Trans. Math. Software, to appear.

[5] C. H. Bischof and G. M. Shroff, On updating signal subspaces, IEEE Trans. Signal Process., SP-40 (1992), pp. 96–105.

[6] Å. Björck, *A bidiagonalization algorithm for solving ill-posed systems of linear equations*, BIT, 28 (1998), pp. 659–670.

[7] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[8] T. F. Chan, *Rank-revealing QR factorizations*, Lin. Alg. Appl., 88/89 (1987), pp. 67–82.

[9] T. F. Chan and P. C. Hansen, *Low-rank revealing QR factorizations*, Numer. Lin. Alg. Appl., 1 (1994), pp. 33–44.

[10] S. Chandrasekaran and I. Ipsen, *On rank-revealing QR factorizations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 592–622.

[11] A. K. Cline, A. R. Conn, and C. F. Van Loan, *Generalizing the LINPACK condition estimator;* in J. P. Hennart (Ed.), Numerical Analysis, Lecture Notes in Mathematics, Vol. 909, Springer, Berlin, 1992.

[12] L. Eldén and E. Sjöström, *Fast computation of the principal singular vectors of Toeplitz matrices arising in exponential data modelling*, Signal Proc., 50 (1996), pp. 151–164.

[13] R. D. Fierro, *Perturbation analysis for two-sided (or complete) orthogonal decompositions*, SIAM J. Matrix Anal. Appl, 17 (1996), pp. 383–400.

[14] R. D. Fierro and J. R. Bunch, *Bounding the subspaces from rank revealing two-sided orthogonal decompositions*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 743–759.

[15] R. D. Fierro and P. C. Hansen, *Low-rank revealing two-sided orthogonal decompositions*, Numer. Algo., 15 (1997), pp. 37–55.

[16] R. D. Fierro, P. C. Hansen, and P. S. K. Hansen, *UTV Tools: Matlab templates for rank-revealing UTV decompositions*, Numer. Algo., 20 (1999), pp. 165–194.

[17] L. V. Foster, *Rank and null space calculations using matrix decomposition without column interchanges*, Lin. Alg. Appl., 74 (1986), pp. 47–71.

[18] M. Gu and S. Eisenstat, *Efficient algorithms for computing strong rank-revealing QR factorizations*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.

[19] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.

[20] G. Heinig and A. Boyanczyk, *Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices II. Algorithms*, Lin. Alg. Appl., 278 (1998), pp. 11–36.

[21] N. J. Higham, *Analysis of the Cholesky decomposition of a semi-definite matrix;* in M. G. Cox and S. J. Hammarling (Eds.), *Reliable Numerical Computing*, Oxford University Press,

1990.
[22] Y. P. Hong and C.-T. Pan, *The rank revealing QR decomposition and SVD*, Math. Comp., 58 (1992), pp. 213–232.
[23] F. T. Luk and S. Qiao, *A symmetric rank-revealing Toeplitz matrix decomposition*, J. VLSI Signal. Proc., 14 (1996), pp. 19–28.
[24] H. Park and L. Eldén, *Downdating the rank revealing URV decomposition*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 138–155.
[25] D. J. Pierce, *A Sparse URL Rather Than a URV Factorization*, Report MEA-TR-203, Boeing Computer Services, Seattle, Washington.
[26] D. J. Pierce and J. G Lewis, *Sparse multifrontal rank revealing QR factorization*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 159–180.
[27] G. Quintana-Ortí and E. S. Quintana-Ortí, *Parallel codes for computing the numerical rank*, Lin. Alg. Appl., 275–276 (1998), pp. 451–470.
[28] I. Slapničar, *Componentwise analysis of direct factorization of real symmetric and Hermitian matrices*, Lin. Alg. Appl., 272 (1998), pp. 227–275.
[29] M. Stewart, *Cholesky factorization of semi-definite Toeplitz matrices*, Lin. Alg. Appl., 254 (1997), pp. 497–526.
[30] G. W. Stewart, *An updating algorithm for subspace tracking*, IEEE Trans. Signal Proc., 40 (1992), pp. 1535–1541.
[31] G. W. Stewart, *Updating a rank-revealing ULV decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), 494–499.