Computing the rank revealing factorization of symmetric matrices by the semiseparable reduction

N. Mastronardi   M. Van Barel   R. Vandebril

Report TW 418, March 2005

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Computing the rank revealing factorization of symmetric matrices by the semiseparable reduction

N. Mastronardi *M. Van Barel* *R. Vandebril*

Report TW418, March 2005

Department of Computer Science, K.U.Leuven

Abstract

An algorithm for reducing a symmetric dense matrix into a symmetric semiseparable one by orthogonal similarity transformations and an efficient implementation of the QR–method for symmetric semiseparable matrices have been recently proposed. In this paper, exploiting the properties of the latter algorithms, an algorithm for computing the rank revealing factorization of symmetric matrices is constructed.

Keywords : rank–revealing decomposition, matrix approximation, symmetric matrices, semiseparable matrices, Lanczos algorithm.
AMS(MOS) Classification : Primary : 65F15, Secondary : 65F25.

COMPUTING THE RANK REVEALING FACTORIZATION OF SYMMETRIC MATRICES BY THE SEMISEPARABLE REDUCTION*

NICOLA MASTRONARDI†, MARC VAN BAREL‡, AND RAF VANDEBRIL§

Abstract. An algorithm for reducing a symmetric dense matrix into a symmetric semiseparable one by orthogonal similarity transformations and an efficient implementation of the QR–method for symmetric semiseparable matrices have been recently proposed. In this paper, exploiting the properties of the latter algorithms, an algorithm for computing the rank revealing factorization of symmetric matrices is constructed.

Key words. rank–revealing decomposition, matrix approximation, symmetric matrices, semiseparable matrices, Lanczos algorithm

AMS subject classifications.

1. Introduction.

The computation of the symmetric rank revealing factorization of dense symmetric matrices is an important problem in signal processing, where accurate computation of the numerical rank, as well as the numerical range and the numerical null space, is required [3, 7, 1, 2].

A family of algorithms for computing symmetric rank revealing decompositions has been presented in [3, 7]. The choice of the algorithm depends on the definiteness of the matrix and the indefinite case seems to be the most difficult to handle.

An algorithm for reducing a symmetric dense matrix into a symmetric semiseparable one by an orthogonal similarity transformation has been recently described [12]. Moreover, an efficient implementation of the QR–method for symmetric semiseparable matrices has been recently proposed [14]. In this paper, exploiting the properties of the latter algorithms, we propose an algorithm for computing the rank revealing decomposition of symmetric matrices.

The paper is organized as follows. In § 2 the definition and properties of a symmetric rank revealing factorization are stated. In § 3, the reduction of a symmetric matrix into a similar semiseparable one, as proposed in [12], is shortly described. Moreover, the relationships of the latter algorithm with the Householder tridiagonalization, the Lanczos algorithm and the QR–method for computing the eigenvalues of a symmetric matrix are exploited. The rank reduction decomposition based on semiseparable matrices is described in § 4. Numerical tests can be found in § 5, followed by the conclusions and future work.

*This research was partially supported by by MIUR, grant number 2004015437 (first author), and by the Research Council K.U.Leuven, project OT/00/16 (SLAP: Structured Linear Algebra Package), by the Fund for Scientific Research–Flanders (Belgium), projects G.0078.01 (SMA: Structured Matrices and their Applications), G.0176.02 (ANCILA: Asymptotic aNalysis of the Convergence behavior of Iterative methods in numerical Linear Algebra), G.0184.02 (CORFU: Constructive study of Orthogonal Functions) and G.0455.0 (RHPH: Riemann-Hilbert problems, random matrices and Padé-Hermite approximation), and by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, project IUAP V-22 (Dynamical Systems and Control: Computation, Identification & Modelling) (second and third authors). The scientific responsibility rests with the authors.

† Istituto per le Applicazioni del Calcolo "M. Picone",sez. Bari, Consiglio Nazionale delle Ricerche Via G. Amendola 122/D, I-70126 Bari, Italy.(n.mastronardi@ba.iac.cnr.it).

‡Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200 A, 3001 Heverlee (Leuven), Belgium. (marc.vanbarel@cs.kuleuven.ac.be).

§Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200 A, 3001 Heverlee (Leuven), Belgium. (raf.vandebril@cs.kuleuven.ac.be).

2. Symmetric rank revealing factorization. A rank revealing decomposition is a decomposition in which information about the *numerical rank* of the matrix can be easily extracted. In this context, numerical rank means the number of singular values of the matrix greater than a certain threshold τ_1. In other words, there has to be a well–determined gap in the singular value distribution at the threshold τ_1 [4], [6, sec. 3.1]. Indeed, rank revealing decompositions are also called *gap revealing decompositions* [10].

The symmetric rank revealing (SRR) decomposition of a matrix

$$A = W \Lambda W^T = \sum_{i=1}^{n} \lambda_i w_i w_i^T,$$

where λ_i and $w_i,\ i = 1, \ldots, n$ are the eigenvalues and the corresponding eigenvectors of A, respectively, with $|\lambda_1| = \sigma_1 \geq |\lambda_2| = \sigma_2 \geq \cdots |\lambda_n| = \sigma_n,\ \{\sigma_i\}_{i=1}^n$ the singular values of A, is defined in the following way[1] [7],

$$A = W_C C W_C^T,$$

where

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{12}^T & C_{22} \end{pmatrix},$$

with $C_{11} \in \mathbb{R}^{k \times k}$, W_C an orthogonal matrix, $\mathrm{cond}(C_{11}) \approx \sigma_1/\sigma_k,\ \|C_{12}\|_F^2 + \|C_{22}\|_F^2 \approx \sigma_{k+1}^2 + \cdots \sigma_n^2$, with $\| \cdot \|_F$ the Frobenius norm.

Of course, if the singular value decomposition (SVD) of A is known, the SRR decomposition is readily obtained. Unfortunately, the SVD is expensive to compute if A is quite large. Hence, it is suitable to derive cheaper methods for computing the SRR decomposition.

We show now that the reduction of a symmetric matrix in semiseparable form, described in [12], can be used as a preprocessing step of a SRR algorithm. In many cases this step is sufficient. If not, few steps of the QR–method without shift can be efficiently applied to the computed symmetric similar semiseparable matrix in order to reveal the numerical rank.

3. Householder reduction, Lanczos algorithm and Semiseparable reduction. In this section, the Householder reduction of A into a symmetric tridiagonal one and the reduction of A into a symmetric semiseparable one by orthogonal similarity transformations are shortly described (more details can be found, e.g., in [5, 12]). Moreover, the relationships of the latter reductions with the Lanczos algorithm [5] are exploited.

Given a vector $x \in \mathbb{R}^l$, $l \in \mathbb{N}$, we denote by `householder` the function

$$[H] = \texttt{householder}(x), \text{ such that } Hx = \alpha e_{1,l}$$

where $\alpha = \mp \|x\|_2$ and $e_{1,l}$ denotes the first vector of the canonical basis of $\mathbb{R}^l$, and $H = I_l - \beta w w^T$ is the Householder matrix, with $w = x \pm \|x\|_2 e_{1,l}$, $\beta = 1/\|w\|_2^2$ and I_l is the identity matrix of order l [5].

[1] This definition is similar to the one used in [8], except that $\|\mathrm{triu}(C_{22})\|_F^2$ instead of $\|C_{22}\|_F^2$ is used in [8].

Moreover, given a vector $y = [y_1, y_2]^T \in \mathbb{R}^2$, we denote by `givens` the function

$$[G] = \texttt{givens}(y), \;\text{ such that }\; Gy = \gamma e_{1,2},$$

where $\gamma = \|y\|_2$ and G is the orthogonal Givens matrix

$$G = \begin{pmatrix} c & s \\ -s & c \end{pmatrix},$$

with $c = y_1/\|y\|_2$ and $s = y_2/\|y\|_2$.

3.1. Householder reduction. The reduction of a symmetric matrix into a symmetric tridiagonal one by means of orthogonal similarity transformations can be accomplished in $n-2$ iterations. At each iteration one more row and column are reduced to the desired tridiagonal structure. The Householder reduction can be summarized in the following procedure:

Algorithm 1: Householder tridiagonalization

Input: the symmetric matrix $A \in \mathbb{R}^{n\times n}$
Output: a similar symmetric tridiagonal matrix $T^{(n-1)}$
$T^{(1)} = A$;
`for i=1:n-2,`
  $\tilde{H}^{(i)} = \texttt{householder}(T^{(i)}(i+1:n, i))$; $H^{(i)} = \texttt{diag}(I_i, \tilde{H}^{(i)})$;
  $T^{(i+1)} = H^{(i)} T^{(i)} {H^{(i)}}^T$;
`end`

Let $\hat{T}^{(i)} = T^{(i)}(1:i, 1:i), \mathcal{H}^{(i)} = H^{(i)} H^{(i-1)} \cdots H^{(1)}$, and $\hat{\mathcal{H}}^{(i)} = \mathcal{H}^{(i)}(1:i+1, :)^T$. The submatrices $\hat{T}^{(i)}$ have already the final tridiagonal structure and $T^{(i)}$ has the following structure,

$$T^{(i)} = \left(\begin{array}{c|ccc} \hat{T}^{(i)} & & & \\ & \times & \cdots & \times \\ \hline \times & \times & \cdots & \times \\ \vdots & \vdots & \ddots & \vdots \\ \times & \times & \cdots & \times \end{array}\right) = \left(\begin{array}{cccc|ccc} \alpha_1 & \beta_1 & & & & & \\ \beta_1 & \ddots & \ddots & & & & \\ & \ddots & \ddots & \beta_{i-1} & & & \\ & & \beta_{i-1} & \alpha_i & \times & \cdots & \times \\ \hline & & & \times & \times & \cdots & \times \\ & & & \vdots & \vdots & \ddots & \vdots \\ & & & \times & \times & \cdots & \times \end{array}\right).$$

A remarkable feature of the latter reduction is the equivalence to the Lanczos' procedure for reducing a symmetric matrix into a similar symmetric tridiagonal one choosing as initial guess the vector $e_{1,n}$ (see [5, p. 472–473]). Therefore,

$$\text{span}(\hat{\mathcal{H}}^{(i)}) = \text{span}(e_{1,n}, Ae_{1,n}, \ldots, A^{i-1}e_{1,n}, A^i e_{1,n}). \tag{3.1}$$

3.2. Reduction to semiseparable form. An algorithm for reducing a symmetric matrix into a similar symmetric semiseparable one, has been presented in [12]. Before considering it, let us introduce the definition of semiseparable matrices (see [15] for more details).

DEFINITION 3.1. *A matrix S is semiseparable if the rank of every submatrix of the lower triangular part (main diagonal included) is less or equal to one (similar for the upper part).*

Therefore, a symmetric semiseparable matrix has the following structure,

$$S = \operatorname{diag}(S_1, \ldots, S_k),$$

where

$$S_i = \begin{pmatrix} v_1u_1 & v_1u_2 & v_1u_3 & \cdots & v_1u_{n_i} \\ v_1u_2 & v_2u_2 & v_2u_3 & \cdots & v_2u_{n_i} \\ v_1u_3 & v_2u_3 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ v_1u_{n_i} & v_2u_{n_i} & \cdots & \cdots & v_{n_i}u_{n_i} \end{pmatrix}, \quad i = 1, \ldots, k,$$

i.e., the lower triangular part of S_i is equal to the lower triangular part of the rank–one matrix $\mathbf{u}_i\mathbf{v}_i^T$ (symmetrically, the upper triangular part is equal to the upper triangular part of $\mathbf{v}_i\mathbf{u}_i^T$), with $\mathbf{u}_i = [u_1, \ldots, u_{n_i}]^T$ and $\mathbf{v}_i = [v_1, \ldots, v_{n_i}]^T, i = 1, \ldots, k,\ k \geq 1$.

Without loss of generality, in the sequel we assume $k = 1$.

REMARK 3.1. *A comprehensive treatment of* semiseparable–like *matrices can be found in [13]. A more stable end efficient representation of semiseparable matrices from a computational point of view, is considered in section 4.*

At the beginning of the ith iteration of the algorithm described in [12], the first i rows and columns are already in the semiseparable form. At the end of the latter iteration the first $i + 1$ rows and columns have the desired semiseparable structure. The algorithm is summarized as follows.

Algorithm 2: semiseparable reduction

Input: the symmetric matrix $A \in \mathbb{R}^{n\times n}$
Output: a similar symmetric semiseparable matrix $S_0^{(n-1)}$

$S_0^{(1)} = A$;
`for i=1:n-2,`
    `% Step i.1`
    $\hat{V}^{(i)} = \texttt{householder}(S_0^{(i)}(i+1:n, i));\quad V^{(i)} = \texttt{diag}(I_i, \tilde{V}^{(i)})$;
    $S_1^{(i)} = V^{(i)} S_0^{(i)} {V^{(i)}}^T$;
    `% Step i.2`
    `for j=i:-1:1,`
        $k = i - j + 1$;
        $\hat{G}_k^{(i)} = \texttt{givens}(S_k^{(i)}(j,j), S_k^{(i)}(j+1, j));\quad G_k^{(i)} = \texttt{diag}(I_{j-1}, \hat{G}_k^{(i)}, I_{n-j-1})$;
        $S_{k+1}^{(i)} = G_k^{(i)} S_k^{(i)} {G_k^{(i)}}^T$;
    `end`
    $S_0^{(i+1)} = S_{i+1}^{(i)}$;
`end`

The graphical description of the 4–th iteration of the latter algorithm for a symmetric matrix of order 8, is depicted in Fig. 3.1. The symbol "⊠" denotes the entries of the matrix sharing already the semiseparararable structure. Moreover, the symbol "⊗" denotes the entries of the matrix to be annihilated by the considered orthogonal similarity transformation and the arrows indicate the rows and columns on which the orthogonal transformation acts.

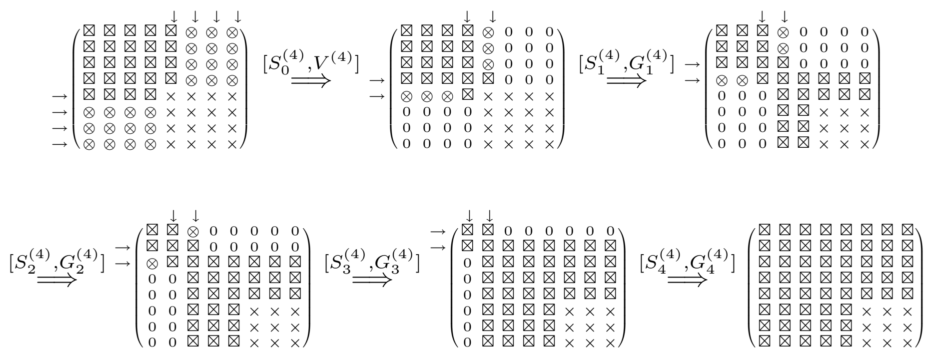

FIG. 3.1. *Description of the similarity transformations used at the 4–th iteration of the algorithm for reducing a symmetric matrix of order 8 into a semiseparable one. The notation $[C, Q]$, with C and Q square matrices and Q orthogonal, indicates the orthogonal similarity transformation QCQ^T.*

Denote by

$$Q^{(i)} = G_i^{(i)} G_{i-1}^{(i)} \cdots G_1^{(i)},\ i = 1, \ldots, n-2. \tag{3.2}$$

It turns out that the latter matrices have the following structure

$$Q^{(i)} = \begin{pmatrix} \hat{Q}^{(i)} & \\ & I_{n-i-1} \end{pmatrix},\ \ i = 1, \ldots, n-2,$$

where $\hat{Q}^{(i)} \in \mathbb{R}^{i+1}$ are upper Hessenberg orthogonal matrices [5]. Due to the semiseparable structure of the matrices $S_1^{(i)}(1 : i+1, 1 : i+1)$, it can be easily checked that the orthogonal matrices $\hat{Q}^{(i)}$ are the Q factors of the QR–factorizations of the latter matrices (see Fig. 2). Therefore, at `Step i.2` of Algorithm 2, the matrix $S_{i+1}^{(i)}(1 : i+1, 1 : i+1) = S_0^{(i+1)}(1 : i+1, 1 : i+1)$, is computed applying one iteration of the QR–algorithm without shift [9] to the matrix $S_1^{(i)}(1 : i+1, 1 : i+1)$ (see Fig. 3.2). Summarizing, at the ith iteration, after having applied the similarity Householder transformation $S_1^{(i)} = V^{(i)} S_0^{(i)} {V^{(i)}}^T$, one more row and column is added to the leading semiseparable submatrix of order i and one step of the QR–algorithm without shift is applied to the augmented leading submatrix of order $i+1$ to reduce it into a semiseparable one. This makes the entries of the leading submatrix of order $i+1$ outside the main diagonal smaller and smaller and the entries on the main diagonal decreasingly ordered in magnitude [16]. The latter behavior is suitable for a rank revealing decomposition.

Checking the structure of the matrices $Q^{(j)}$ and $V^{(j)}$, $j = 1 \ldots, n-2$, it turns out

$$V^{(l)} Q^{(m)} = Q^{(m)} V^{(l)},\ \ m < l \leq n-2.$$

Therefore, the matrices $S_0^{(i+1)}$, $i = 1, \ldots, n-2$, can be written in the following way.

$$\begin{aligned} S_0^{(i+1)} &= Q^{(i)} V^{(i)} Q^{(i-1)} V^{(i-1)} \cdots Q^{(1)} V^{(1)} A {V^{(1)}}^T {Q^{(1)}}^T \cdots {V^{(i-1)}}^T {Q^{(i-1)}}^T {V^{(i)}}^T {Q^{(1)}}^T \\ &= Q^{(i)} \cdots Q^{(1)} V^{(i)} \cdots V^{(1)} A {V^{(1)}}^T \cdots {V^{(i)}}^T {Q^{(1)}}^T \cdots {Q^{(i)}}^T. \end{aligned}$$

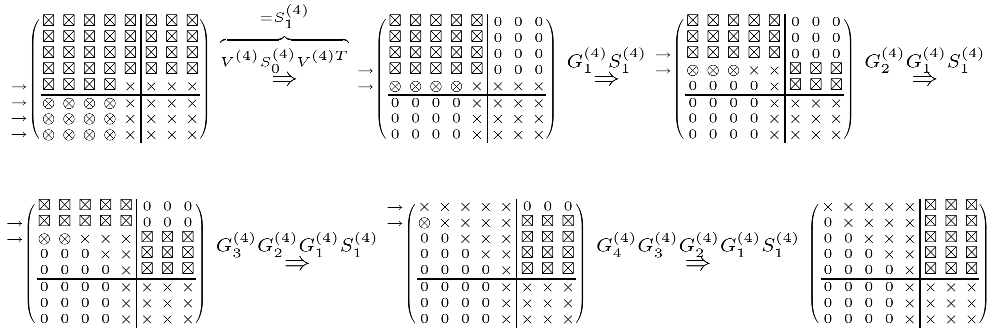

FIG. 3.2. *Applying the orthogonal transformations only to the left side of the matrix in the Step i.2 of Algorithm 2, the leading submatrix of order $i+1$ becomes upper triangular.*

The following relations are straightforward to derive.

$$H^{(k)} = V^{(k)}, \qquad k = 1, \ldots, n-2. \tag{3.3}$$

$$\lambda(T^{(i)}(1:i,1:i)) = \lambda(S_0^{(i)}(1:i,1:i)),\ i = 1, \ldots, n-1, \tag{3.4}$$

where $\lambda(C)$ denotes the spectrum of the square matrix C.

By (3.4), the submatrices $\hat{T}^{(i)}$ and $S_0^{(i)}(1:i,1:i)$ generated by the Householder reduction and by the semiseparable reduction, respectively, share the same eigenvalues. Therefore, $\|\hat{T}^{(i)}\|_F = \|S_0^{(i)}(1:i,1:i)\|_F$. We recall that the eigenvalues of $\hat{T}^{(i)}$ are the Ritz values of the Lanczos process applied to A with starting vector $e_{1,n}$.

Let $\mathcal{Q}^{(i)} = Q^{(i)}V^{(i)}Q^{(i-1)}V^{(i-1)} \cdots Q^{(1)}V^{(1)}$ and $\hat{\mathcal{Q}}^{(i)} = \mathcal{Q}^{(i)}(1:i+1,:)^T$. Taking (3.1) and (3.3) into account, it turns out

$$\mathrm{span}(\hat{\mathcal{Q}}^{(i)}) = \mathrm{span}(e_{1,n}, Ae_{1,n}, \ldots, A^{i-1}e_{1,n}, A^i e_{1,n}).$$

Moreover,

$$\beta_i = \|S_0^{(i)}(i+1:n,1:i)\|_F = \|S_1^{(i)}(i+1,1:i)\|_2,\ \ i = 1, \ldots, n-1. \tag{3.5}$$

4. Symmetric Rank Revealing decomposition via semiseparable reduction. An efficient implementation of the QR–method without shift for computing the spectral decomposition of symmetric semiseparable matrices has been recently proposed in [14]. It requires $O(n)$ flops and $O(n)$ memory per step, with n the order of the matrix. The proposed rank revealing factorization has the reduction of the original matrix into a symmetric semiseparable one as preprocessing step, followed by few steps of the QR–method without shift, if needed.

In this section we describe the so called *Givens–vector* (*G–d*) representation for semiseparable matrices and how useful information for the implementation of a symmetric rank–revealing algorithm, based on the reduction into semiseparable matrices, can be retrieved from the latter representation. The details of the efficient implementation of the QR–method without shift for computing the spectral decomposition of a symmetric semiseparable matrix can be found in [14].

For the sake of clarity, let us introduce the following notations.

$$\begin{cases} s_{j:i} \equiv \prod_{k=j}^{i} s_j, & j \leq i, \\ s_{j:i} \equiv 1, & j > i. \end{cases}$$

There are different ways of representing a semiseparable matrix. A comprehensive description of representations of semiseparable matrices can be found in [15]. A useful representation, from a computational point of view, is the *G–d* representation.

A symmetric semiseparable matrix S can always be written in the following form,

$$S = \begin{pmatrix} c_1 d_1 & c_2 s_1 d_1 & c_3 s_{1:2} d_1 & \cdots & c_{n-1} s_{1:n-2} d_1 & s_{n-1} s_{1:n-2} d_1 \\ c_2 s_1 d_1 & c_2 d_2 & c_3 s_2 d_2 & \cdots & c_{n-1} s_{2:n-2} d_1 & s_{n-1} s_{2:n-2} d_1 \\ c_3 s_{1:2} d_1 & c_3 s_2 d_2 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & c_{n-2} d_{n-2} & c_{n-1} s_{n-2} d_{n-2} & s_{n-1} s_{n-2} d_{n-2} \\ c_{n-1} s_{1:n-2} d_1 & c_{n-1} s_{2:n-2} d_1 & \cdots & c_{n-1} s_{n-2} d_{n-2} & c_{n-1} d_{n-1} & s_{n-1} d_{n-1} \\ s_{n-1} s_{1:n-2} d_1 & s_{n-1} s_{2:n-2} d_1 & \cdots & s_{n-1} s_{n-2} d_{n-2} & s_{n-1} d_{n-1} & d_n \end{pmatrix},$$

where c_i and s_i, $i = 1, \ldots, n-1$, are suitable Givens coefficients. It turns out that S can be represented by a vector $\mathbf{d} = (d_1, d_2, \ldots, d_n)^T$, and a $2 \times (n-1)$ matrix

$$G = \begin{pmatrix} c_1 & c_2 & \ldots & c_{n-1} \\ s_1 & s_2 & \ldots & s_{n-1} \end{pmatrix}.$$

In this case, the symmetric semiseparable matrix is said to be represented in *G–d* form.

The diagonal entries of a symmetric semiseparable matrix S represented in *G–d* form can be easily retrieved. Indeed,

$$\begin{cases} S(i,i) = c_i d_i, & i = 1, \ldots, n-1, \\ S(i,i) = d_i & i = n. \end{cases} \tag{4.1}$$

Moreover, the following lemma holds.

Lemma 4.1. *Let S be a symmetric semiseparable matrix of order n represented in G–d form. Then*

$$\|S(i+1:n, 1:i)\|_W = |s_i| \sqrt{\|S(i:n, 1:i-1)\|_W^2 + d_i^2}, \; i = 1, \ldots, n-1, \tag{4.2}$$

where $\| \cdot \|_W$ can be either the spectral or the Frobenius norm.

Proof. The submatrices $S(i+1:n, 1:i)$ have rank one, i.e.,

$$S(i+1:n, 1:i) = \mathbf{u}_i \mathbf{v}_i^T,$$

with

$$\mathbf{u}_i = \begin{pmatrix} c_{i+1} & c_{i+2} s_{i+1} & c_{i+3} s_{i+1:i+2} & \cdots & c_{n-1} s_{i+1:n-2} & s_{n-1} s_{i+1:n-2} \end{pmatrix}^T,$$
$$\mathbf{v}_i = \begin{pmatrix} s_{1:i} d_1 & s_{2:i} d_2 & \cdots & s_{i-1:i} d_{i-1} & s_i d_i \end{pmatrix}^T.$$

It turns out that

$$\|\mathbf{u}_i\|_2^2 = \underbrace{c_{i+1}^2 + \underbrace{c_{i+2}^2 s_{i+1}^2 + \underbrace{c_{i+3}^2 s_{i+1:i+2}^2 + \cdots + \underbrace{c_{n-1}^2 s_{i+1:n-2}^2 + s_{n-1}^2 s_{i+1:n-2}^2}_{=s_{i+1:n-2}^2}}_{=s_{i+1:i+2}^2}}_{=s_{i+1}^2}}_{=1}$$

Therefore $\|S(i+1:n,1:i)\|_2 = \|\mathbf{v}_i\|_2$.
On the other hand,

$$\begin{aligned}\|\mathbf{v}_i\|_2^2 &= \sum_{j=1}^{i} d_j^2 \prod_{k=j}^{i} s_j^2 \\ &= d_i^2 s_i^2 + s_i^2 \sum_{j=1}^{i-1} d_j^2 \prod_{k=j}^{i-1} s_j^2 \\ &= s_i^2 \left(d_i^2 + \|\mathbf{v}_{i-1}\|_2^2 \right).\end{aligned}$$

□

Formulae (4.1) and (4.2) can be computed recursively. If one is interested to compute them at each step of the reduction of Algorithm 2, the cost is linear with the size of the already computed semiseparable matrix. Otherwise they can be computed after a fixed number of steps of the reduction.

With similar arguments, it is possible to show that the diagonal entries of $S^T S$ can be computed with the following recursive procedure (in case the order of the semiseparable matrix is less than 3, the computation is trivial).

$\hat{d}_1 = d_1^2$;
$t = s_1^2 \hat{d}_1$
$\hat{d}_2 = c_2^2 t + d_2^2$;
$t = t + d_2^2$;
`for` $i = 3 : n-1$,
    $t = s_{i-1}^2 t$;
    $\hat{d}_i = c_i^2 t + d_i^2$;
    $t = t + d_i^2$;
    `if` $i = n-1$,
        $\hat{d}_n = s_{n-1}^2 t + d_n^2$;
    `end`
`end`

Therefore, the Frobenius norm of a symmetric semiseparable matrix S, expressed in G–d form, can be easily retrieved with linear complexity considering the square root of the sum of the diagonal elements of $S^T S$.

The rank–revealing algorithm based on the reduction into semiseparable matrices, can now be formulated.

Algorithm 3: Rank–Revealing Semiseparable (RRS) factorization

Input: the symmetric matrix $A \in \mathbb{R}^{n \times n}$
τ_1 : the threshold for the rank revealing
τ_2 : the threshold for interrupting the reduction to semiseparable matrix
Output: the RRS factorization of A
1) compute the Frobenius norm of A
2) compute the sequence of the matrices $S_0^{(i)}$, $i = 1, \ldots, n-1$, as in Algorithm 2 until $\|A\|_F - \|S_0^{(j+1)}(1:j+1, 1:j+1)\|_F < \tau_2$, with $1 \leq j \leq n-1$.
3) Define $S^{[0]} \equiv S_0^{(j+1)}(1:j+1, 1:j+1)$.
4) apply few steps, let us say l steps, of the QR–method without shift to the semiseparable matrix $S^{[0]}$:

$[Q, R] = qr(S^{[i]}),\ \ S^{[i+1]} = RQ,\ \ i = 0, 1, \ldots, l-1.$

5) compute $\|S^{[l]}(k+1 : j+1, 1 : k)\|_F,\ k = j, j-1, \ldots, 1,$
and $S^{[l]}(k, k),\ k = 1, \ldots, j+1$, in order "to deflate" the matrix $S^{[l]}$ removing the singular values that are below the threshold τ_1.

Step 2) of algorithm 3 is the most expensive one. Indeed, the computational complexity is $O(jn^2 + j^3)$. Moreover, it requires $O(jn)$ memory to store the involved Householder vectors and the Givens coefficients. As already said, each iteration of the QR–method without shift in step 4) requires $O(j)$ computational complexity and storage. The choice of l depends on the handled matrix. Often l can be chosen very small compared to the size of the matrix. Step 5) is also computed in $O(j)$ flops. The choice of threshold τ_2 is crucial for the accuracy of the results. In all the examples of the next section, $\tau_2 = n\epsilon\|A\|_F$, where n is the order of the matrix, $\|\ \cdot\ \|_F$ is the Frobenius norm, and $\epsilon \sim 2.2204\text{e-}016$ is the machine precision. With this choice of the threshold τ_2, the reduction to semiseparable form is entirely performed, for all the considered examples.

It is well known that the single–vector Lanczos algorithm can compute multiple eigenvalues of a symmetric matrix, but the multiple eigenvalues do not necessarily converge consecutively one after another. More precisely, if $\lambda_{\max}(A)$ is a multiple eigenvalue with maximum modulus of a symmetric matrix A, then usually a copy of $\lambda_{\max}(A)$ will converge first, followed by several other smaller eigenvalues of A. Then another copy of $\lambda_{\max}(A)$ will converge followed again by smaller eigenvalues of A, and so on. The consequence of this convergence behavior is that only fewer copies of the largest eigenvalues of A are also eigenvalues of $T^{(k)}$. The semiseparable reduction behaves similarly to the Lanczos reduction to tridiagonal matrices. Therefore, in case of multiple eigenvalues of A, the symmetric semiseparable matrix computed by the reduction is block diagonal, with each block having $\lambda_{\max}(A)$ as eigenvalue.

A multi–vector semiseparable reduction can be considered in such cases [11].

The effect of steps 4) and 5) of the latter algorithm is to make the entries on the main diagonal decreasingly ordered. Therefore, the smallest eigenvalues appear as the last entries of the main diagonal and the Frobenius norms of the corresponding rows and columns are very small. If the latter norms and the size of the corresponding eigenvalues are below the tolerance τ_1, the algorithm proceeds removing the rows and columns from the matrix.

5. Numerical experiments. Some of the matrices used in [3, 7] are considered to test the proposed rank revealing factorization.

For the matrices considered in the examples, the rank revealing factorizations are computed correctly, and the numerical ranks are always equal to the desired ones. The results of the proposed rank revealing factorization based on semiseparable matrices are compared to the results obtained by using the `matlab` functions `hvsvsd` (semipositive definite case) and `hvsvid_L` (indefinite case) of the `matlab` toolbox UTV [1, 2]. The number of steps of the QR–method without shift in the proposed algorithm is fixed to 10 for all the examples. As in [3, 7], the symmetric semidefinite test matrices are computed as $Q\Sigma Q^T$, with Q random orthogonal matrices and Σ diagonal matrices with the desired singular values. For the indefinite case, the diagonal matrices are replaced by the product of diagonal matrices of the desired singular values by diagonal random sign matrices. The experiments were carried out on a PC with `matlab 5.3` to compute the number of flops. It can be noticed that the proposed algorithm is the fastest one. Moreover, it is as accurate as those available in the literature.

Positive definite case

n	max $\lvert\kappa_2(\hat{A}) - \kappa_2(H_{11})\rvert$	max $\lvert\kappa_2(\hat{A}) - \kappa_2(S_{11})\rvert$	max $\lVert H_{12}\rVert_F$	max $\lVert S_{12}\rVert_F$	average $\#flops^{(H)}$	average $\#flops^{(S)}$
64	6.88e-09	1.00e-08	8.33e-11	2.14e-38	6.0390e+06	2.7410e+06
128	4.49e-09	1.17e-08	1.22e-10	1.66e-37	5.9928e+05	2.4800e+05
256	8.26e-09	3.48e-08	1.24e-10	3.26e-36	3.8894e+06	1.6242e+06

Indefinite case

n	max $\lvert\kappa_2(\hat{A}) - \kappa_2(H_{11})\rvert$	max $\lvert\kappa_2(\hat{A}) - \kappa_2(S_{11})\rvert$	max $\lVert H_{12}\rVert_F$	max $\lVert S_{12}\rVert_F$	average $\#flops^{(H)}$	average $\#flops^{(S)}$
64	4.86e-09	1.07e-08	5.48e-13	3.18e-39	1.1771e+05	2.7478e+04
128	2.99e-09	2.34e-09	2.66e-13	8.15e-38	1.4552e+06	2.4793e+05
256	3.52e-09	3.74e-09	4.53e-13	2.82e-37	1.0641e+07	2.9243e+06

TABLE 5.1

Results obtained applying the rank revealing algorithms `hvsvsd` *(semipositive definite case),* `hvsvid_L` *(indefinite case) and the proposed one to 100 symmetric random matrices with prescribed singular values, constructed as described in example 5.2.*

EXAMPLE 5.1. *The matrices considered in this example are randomly generated test matrices of size* $n = 64, 128, 256$, *(100 matrices for each size), each with* $n - 4$ *singular values geometrically distributed between* 1 *and* 10^{-4}, *and the remaining* 4 *singular values given by* 10^{-7}, 10^{-8}, 10^{-9}, *and* 10^{-10}, *such that the numerical rank with respect to the threshold* $\tau_1 = 10^{-5}$ *is* $k = n - 4$.

The results are depicted in table 5.1. The size of the matrices is in first column. In column 2, the maximum, over 100 matrices, of the difference between $\kappa_2(\hat{A})$, *the condition number of the leading submatrices of order* $n - k$ *of the symmetric generated matrices and* $\kappa_2(H_{11})$, *the condition number of the leading submatrices of order* $n - k$ *of the rank revealing matrices* H *computed by the function of UTV toolbox [1, 2] is reported and, in column 3, the maximum, over 100 matrices, of the difference between* $\kappa_2(\hat{A})$ *and* $\kappa_2(S_{11})$, *the condition number of the leading submatrices of order* $n - k$ *of the rank revealing matrices* S *computed by the proposed algorithm.*

In column 4 and 5, the maximum of the Frobenius norm of submatrices $H_{12} = H(1 : k, k + 1 : n)$ *and* $S_{12} = S(1 : k, k + 1 : n)$, *respectively, are depicted. Finally, in column 6 and 7 the average number of flops, over the 100 matrices, for the algorithms of UTV toolbox [1, 2] and the proposed algorithm, respectively, can be found.*

EXAMPLE 5.2. *The proposed rank revealing factorization is now applied to compute the numerical rank of* 4 *classes of matrices considered in [3]. The size of the considered matrices is* $n = 20$ *(100 matrices for each class), and the desired numerical rank is* $k = 15$. *The singular values are geometrically distributed between* σ_1 *and* σ_k *and also between* σ_{k+1} *and* σ_n. *Moreover* $\sigma_1 = 1$ *and* $\sigma_n = 10^{-10}$.

The classes of considered matrices are constructed according to table 5.2. The considered positive definite and indefinite symmetric matrices are constructed as in the latter example. The results are reported in table 5.3. In the first column, the class of considered matrices are reported. The legend of column 2 up to 7 is similar to the corresponding one of table 5.1. For all these matrices, the threshold is $\tau_1 = (\sigma_k + \sigma_{k+1})/2$.

6. Conclusions. In this paper we have presented an algorithm for computing the rank revealing factorization of symmetric matrices based on an algorithm for reducing symmetric matrices into semiseparable ones by similarity orthogonal transformations

Cl.	$\sigma_1(A)$	$\sigma_n(A)$	$\sigma_k(S)$	$\sigma_k(S)/\sigma_{k+1}(A)$
1	1.0	$1.0 \cdot 10^{-10}$	$1.0 \cdot 10^{-3}$	2
2	1.0	$1.0 \cdot 10^{-10}$	$1.0 \cdot 10^{-3}$	10
3	1.0	$1.0 \cdot 10^{-10}$	$1.0 \cdot 10^{-6}$	2
4	1.0	$1.0 \cdot 10^{-10}$	$1.0 \cdot 10^{-6}$	10

TABLE 5.2
Features of the singular values of the classes of matrices considered for the example 2.

Positive definite case

Cl.	max $\lvert\kappa_2(\hat{A}) - \kappa_2(H_{11})\rvert$	max $\lvert\kappa_2(\hat{A}) - \kappa_2(S_{11})\rvert$	max $\lVert H_{12}\rVert_F$	max $\lVert S_{12}\rVert_F$	average $\#flops^{(H)}$	average $\#flops^{(S)}$
1	5.70e-11	7.7875e-11	2.25e-04	1.08e-24	6.6502e+04	2.7737e+04
2	6.65e-11	1.4450e-10	9.45e-06	5.34e-23	6.6962e+04	2.7737e+04
3	6.37e-05	6.7613e-05	2.46e-07	2.34e-18	6.6831e+04	2.7737e+04
4	5.97e-05	5.5703e-05	7.44e-09	2.19e-15	6.6962e+04	2.7737e+04

Indefinite case

Cl.	max $\lvert\kappa_2(\hat{A}) - \kappa_2(H_{11})\rvert$	max $\lvert\kappa_2(\hat{A}) - \kappa_2(S_{11})\rvert$	max $\lVert H_{12}\rVert_F$	max $\lVert S_{12}\rVert_F$	average $\#flops^{(H)}$	average $\#flops^{(S)}$
1	9.37e-11	5.85e-11	7.19e-04	1.08e-25	1.2417e+05	2.7737e+04
2	5.34e-11	5.60e-11	2.28e-07	3.19e-24	1.2443e+05	2.7737e+04
3	1.01e-04	4.51e-05	6.27e-07	6.41e-18	1.2708e+05	2.7737e+04
4	4.90e-05	4.66e-05	3.14e-09	2.56e-16	1.2758e+05	2.7737e+04

TABLE 5.3
Results obtained applying the rank revealing algorithms `hvsvsd` *(semipositive definite case),* `hvsvid_L` *(indefinite case) and the proposed one to 100 symmetric random matrices with prescribed singular values, constructed as described in example 2.*

and on few steps of the QR–method without shift applied to the latter matrices. Some numerical experiments are reported confirming the reliability of the proposed method. Further research will consist of extending the proposed rank revealing algorithm for symmetric and non symmetric sparse and structured matrices.

REFERENCES

[1] R. D. FIERRO, P. C. HANSEN, AND P. S. K. HANSEN, *UTV Tools: Matlab templates for rank-revealing UTV decompositions*, Numer. Algorithms, 20 (1999), pp. 165–194.
[2] R. D. FIERRO, P. C. HANSEN, *UTV Expansion Pack: special–purpose rank revealing algorithms*, submitted to Numer. Algorithms.
[3] R. D. FIERRO, P. C. HANSEN, *Truncated VSV solutions to symmetric rank-deficient problems*, BIT 42 (2002), pp. 531–540.
[4] G.H. GOLUB, V.C. KLEMA, G.W. STEWART, *Rank degeneracy and least squares problems*, Technical Report CS-TR-76-559 1976, Stanford University Stanford, CA, USA.
[5] G.H. GOLUB, C. F. VAN LOAN, *Matrix Computations*, Third ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[6] P.C. HANSEN, *Rank–deficient and discrete ill–posed problems: numerical aspects of linear inversion,* SIAM monographs on mathematical modeling and computation, SIAM, Philadelphia, 1997.
[7] P. C. HANSEN, P.Y. YALAMOV, *Computing symmetric rank–revealing decompositions via triangular factorization*, SIAM J. on Matrix Anal. Appl., 23:2 (2002) pp. 443–458.
[8] F. T. LUK AND S. QIAO, *A symmetric rank-revealing Toeplitz matrix decomposition*, J. VLSI

Signal Process., 14 (1996), pp. 19–28.

[9] B.N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[10] G. W. Stewart, *The QLP approximation to the singular value decomposition*, *SIAM J. Sci. Comput.*, 20:4 (1999), pp. 1336–1348.

[11] M. Van Barel, E. Van Camp, N. Mastronardi, *Orthogonal similarity transformation into semiseparable matrices of semiseparability rank k*, Department of Computer Science, K.U.Leuven, Report TW 380, Leuven, Belgium, January, 2004, accepted for publication in *Numer. Linear Algebra Appl.*.

[12] M. Van Barel, R. Vandebril, N. Mastronardi, *An orthogonal similarity reduction of a matrix into semiseparable form*, Department of Computer Science, K.U.Leuven, Report TW 355, Leuven, Belgium, February, 2003, SIAM J. Matrix Anal. Appl., to appear.

[13] R. Vandebril, Semiseparable matrices and the symmetric eigenvalue problem, PhD thesis, Katholieke Universiteit Leuven, 2003.

[14] R. Vandebril, M. Van Barel, N. Mastronardi, *An implicit QR algorithm for symmetric semiseparable matrices*, Department of Computer Science, K.U.Leuven, Report TW 367, Leuven, Belgium, August, 2003, accepted for publication in *Numer. Linear Algebra Appl.*.

[15] R. Vandebril, M. Van Barel, N. Mastronardi, *A note on the representation and definition of semiseparable matrices*, Department of Computer Science, K.U.Leuven, Report TW 368, Leuven, Belgium, 2003, accepted for publication in *Numer. Linear Algebra Appl.*.

[16] J.H. Wilkinson, *The Algebraic Eigenvalue problem,* Oxford University press, New York, 1965.