Quasiseparable Hessenberg reduction of real diagonal plus low rank matrices and applications[☆]

Dario A. Bini[a], Leonardo Robol[b,*]

[a]*Dipartimento di Matematica, Università di Pisa, Italy*
[b]*Scuola Normale Superiore, Pisa, Italy*

Abstract

We present a novel algorithm to perform the Hessenberg reduction of an $n\times n$ matrix A of the form $A = D + UV^*$ where D is diagonal with real entries and U and V are $n \times k$ matrices with $k \leq n$. The algorithm has a cost of $O(n^2k)$ arithmetic operations and is based on the quasiseparable matrix technology. Applications are shown to solving polynomial eigenvalue problems and some numerical experiments are reported in order to analyze the stability of the approach.

AMS classification: 65F15, 15A21, 15A03

Keywords: Quasiseparable matrices, Polynomial eigenvalue problem, Hessenberg form

1. Introduction

Reducing an $n \times n$ matrix to upper Hessenberg form by a unitary similarity transformation is a fundamental step at the basis of most numerical methods for computing matrix eigenvalues. For a general matrix, this reduction has a cost of $O(n^3)$ arithmetic operations (ops), while for matrices having additional structures this cost can be lowered. This happens, for instance, for the class of quasiseparable matrices. We say that a matrix is (k_l, k_u)-quasiseparable if its submatrices contained in the strictly upper triangular part have rank at most k_u and those contained in the strictly lower triangular part have rank at most k_l. For simplicity, if $k_l = k_u = k$ we say that A is k-quasiseparable. Quasiseparable matrices have recently received much attention; for properties of this matrix class we refer the reader to the recent books [5], [6], [12], [13].

In the papers [4] and [8], algorithms are provided to reduce a k-quasiseparable matrix A to upper Hessenberg form H via a sequence of unitary transformations. If A satisfies some additional hypothesis then the Hessenberg matrix obtained this way is still quasiseparable, and the cost of this reduction is $O(n^2k^\alpha)$ ops, where α is strictly greater than 1, in particular $\alpha = 3$ for the algorithm of [4] and $\alpha > 1$ in [8], but its value is not exactly deducible as it depends on some choices in the implementation. The advantage of this property is that one can apply the shifted QR iteration to the matrix H at the cost of $O(nk^2)$ ops per step, instead of $O(n^2)$, by exploiting both the Hessenberg and the quasiseparable structure of H, see for instance [7], [11]. This way, the computation of the eigenvalues of A has a lower cost with respect to the case of a general matrix. More specifically, assuming that the number of QR iterations is $O(n)$, the overall cost for computing all the eigenvalues turns to $O(n^2k^2 + n^2k^\alpha)$. Clearly, if $\alpha \geq 3$, the

[☆] Work supported by Gruppo Nazionale di Calcolo Scientifico (GNCS) of INdAM

[*] Corresponding author.

Email addresses: bini@dm.unipi.it (Dario A. Bini), leonardo.robol@sns.it (Leonardo Robol)

advantage of this quasiseparable reduction to Hessenberg form can be appreciated only if $k \ll n$. In particular, if the value of k is of the order of $n^{\frac{1}{3}}$ these algorithms might have a cost of $O(n^3)$ ops as for a general unstructured matrix.

In this paper we consider the case of a k-quasiseparable matrix which can be represented as

$$A = D + UV^*, \quad U, V \in \mathbb{C}^{n\times k}, \tag{1}$$

where D is a diagonal matrix with real entries and $U, V \in \mathbb{C}^{n\times k}$ and V^* is the Hermitian transpose of V. Matrices of this kind are encountered in the linearization of matrix polynomials obtained by the generalization of the Smith companion form [9], [2]. They have been used also in the package MPSolve v. 3.1.4 for the solution of polynomial and secular equations up to any desired precision [3].

We prove that if $H = QAQ^*$ is the Hessenberg form of A, with Q unitary, then H is $(1, 2k-1)$-quasiseparable, moreover, we provide an algorithm for computing H with $O(n^2k)$ ops. This algorithm substantially improves the algorithms of [4] and [8] whatever is the value of k. Moreover, for an unstructured matrix where $k = n$, the cost of our algorithm amounts to $O(n^3)$, that is the same asymptotic cost of the Hessenberg reduction for a general matrix.

It interesting to observe that the improvement of our algorithm over [4] and [8] is due to two different reasons: the quasiseparable structure used in [4] relies on auxiliary *unstructured* $k \times k$ matrices whose manipulation costs $O(k^3)$ ops; in [8] the rank of the quasiseparable structure grows during the computation and its growth is controlled by a compression step which might be expensive.

An immediate consequence of this algorithm is that the cost for computing all the eigenvalues of A by means of the shifted QR iteration applied to H turns to $O(n^2k)$ with an acceleration by a factor of $O(k^{\alpha-1})$ with respect to the algorithms of [4] and [8].

Another application of this result concerns the solution of polynomial eigenvalue problems and is the main motivation of this work. Consider the matrix polynomial $P(x) = \sum_{i=0}^{n} P_i x^i$, $P_i \in \mathbb{C}^{m\times m}$, where for simplicity we assume $P_n = I$. The polynomial eigenvalue problem consists in computing the solution of the equation $\det P(x) = 0$ and, if needed, the nonzero vectors v such that $P(x)v = 0$.

The usual strategy adopted in this case is to employ a linearization $L(x) = xI - A$ of the matrix polynomial, such that the linear eigenvalue problem $L(x)w = 0$ is equivalent to the original one $P(x)v = 0$. The matrix A, also called companion matrix of $P(x)$, is of size $mn \times mn$. Many companion matrices do have a k-quasiseparable structure where $k = m$ and in the case of the Smith companion [9], generalized to matrix polynomials in [2], the quasiseparable structure takes the desired form $A = D + UV^*$. In this case, the cost of our algorithm to reduce A into quasiseparable Hessenberg form turns to $O(nm^2)$ whereas the algorithms of [4] and [8] of cost $O(nm^{\alpha+1})$, where $\alpha > 1$, would be unpractical.

This paper is divided in 6 sections. Besides the introduction, in Section 2 we introduce some preliminary results, concerning quasiseparable matrices, which are needed to design the algorithm for the Hessenberg reduction. In Section 3 we recall the general algorithm for reducing a matrix into Hessenberg form by means of Givens rotations and prove the main result, expressed in Theorem 3.2, on the conservation of the quasiseparable structure at all the steps of the algorithm. In Section 4 we recall and elaborate the definition of Givens vector representation of a symmetric k-quasiseparable matrix. Then we rephrase Theorem 3.2 in terms of Givens vector representations. Section 5 deals with algorithmic issues: the fast algorithm for the Hessenberg reduction is presented in detail. In Section 6 we present some numerical experiments and show that the CPU time needed in our tests confirms the complexity bound $O(n^2k)$. Finally the last Section 7 briefly shows an application of these results to numerically computing the eigenvalues of a matrix polynomial.

2. Preliminary tools

Throughout the paper the matrix A has the form (1) and $H = QAQ^*$ is in upper Hessenberg form, where Q is unitary, i.e., $Q^*Q = I$. Here we recall the notations and definitions used in this paper, which mainly comply with the ones used in [13], together with the main results concerning quasiseparable matrices.

Definition 2.1. A complex matrix A is *lower-quasiseparable* (resp. *upper-quasiseparable*) *of rank* k if every submatrix contained in the strictly lower (resp. upper) triangular part of A has rank at most k. If A is k_l lower quasiseparable and k_u upper quasiseparable we say that A is (k_l, k_u)-quasiseparable. If $k = k_l = k_u$ we say that A is k-quasiseparable. Moreover, we denote $\mathcal{QSH}_k^n$ the set of $n \times n$ Hermitian k-quasiseparable matrices with entries in $\mathbb{C}$. We will sometimes omit the superscript n and simply write $\mathcal{QSH}_k$ when the dimension is clear from the context. In general, we say that A is quasiseparable if it is (k_l, k_u)-quasiseparable for some nontrivial k_l, k_u.

The definition of a quasiseparable matrix can be expressed easily by using the MATLAB notation. In fact, A is lower-quasiseparable of rank k if and only if

$$\operatorname{rank}(A[i+1:n, 1:i]) \leqslant k \qquad \text{for } i = 1, \ldots, n-1,$$

where $A[i_1 : i_2, j_1 : j_2]$ denotes the submatrix of A formed by the entries $a_{i,j}$ for $i = i_1, \ldots, i_2$, $j = j_1, \ldots, j_2$.

Given a vector $v = (v_i) \in \mathbb{C}^2$, denote by $G = G(v_1, v_2)$ a 2×2 Givens rotation

$$G = \begin{bmatrix} c & s \\ -\overline{s} & c \end{bmatrix}, \quad c \in \mathbb{R}, \quad |s|^2 + c^2 = 1,$$

such that $Gv = \alpha e_1$, where $|\alpha| = \sqrt{|v_1|^2 + |v_2|^2}$. More generally, denote $G_i = I_{i-1} \oplus G \oplus I_{n-i-1}$ the $n \times n$ matrix which applies the Givens rotation $G = \left[\begin{smallmatrix} c_i & s_i \\ -\overline{s}_i & c_i \end{smallmatrix}\right]$ to the components i and $i+1$ of a vector, where I_m denotes the identity matrix of size m and $A \oplus B$ denotes the block diagonal matrix $\left[\begin{smallmatrix} A & 0 \\ 0 & B \end{smallmatrix}\right]$. In the following we will call G_i Givens rotations as well.

The following well-known result [13] will be used in our analysis:

Lemma 2.2. *Let Q be a unitary Hessenberg matrix. Then*

- *Q is a $(1,1)$-quasiseparable matrix.*
- *If $\det(Q) = 1$ then Q can be factorized as a product of $n-1$ Givens rotations $Q = G_1 \ldots G_{n-1}$.*

In the following we use the operators $\operatorname{tril}(\cdot,\cdot)$ and $\operatorname{triu}(\cdot,\cdot)$, coherently with the corresponding MATLAB functions, such that $L = \operatorname{tril}(A,k)$, $U = \operatorname{triu}(A,k)$ where $A = (a_{i,j})$, $L = (\ell_{i,j})$, $U = (u_{i,j})$ and

$$\ell_{i,j} = \begin{cases} a_{i,j} & \text{if } i \geqslant j-k \\ 0 & \text{otherwise} \end{cases} \qquad u_{i,j} = \begin{cases} a_{i,j} & \text{if } i \leqslant j-k \\ 0 & \text{otherwise} \end{cases}.$$

2.1. A useful operator

Another useful tool is the operator $t : \mathbb{C}^{n\times n} \to \mathbb{C}^{n \times n}$ defined by

$$t(A) = \operatorname{tril}(A, -1) + \operatorname{triu}(A^*, 1). \tag{2}$$

Observe that if A has rank k then $t(A)$ is a Hermitian k-quasiseparable matrix with zero diagonal entries. In particular, for $u, v \in \mathbb{C}^n$, the matrix $t(uv^*)$ is in $\mathcal{QSH}_1^n$ and its entries are independent of u_1 and of v_n. More generally, for $U, V \in \mathbb{C}^{n\times k}$, the matrix $t(UV^*)$ is in $\mathcal{QSH}_k^n$ and its entries are independent of the first row $U[1,:]$ of U and of the last row $V[n,:]$ of V. Observe also that $t(A)$ is independent of the upper triangular part of A.

The following properties can be verified by a direct inspection

$$\begin{aligned} &t(A+B) = t(A) + t(B), \text{ for any } A, B, \in \mathbb{C}^{n\times n},\\ &t(\alpha A) = \alpha t(A), \text{ for any } \alpha \in \mathbb{R},\\ &t(DAD^*) = Dt(A)D^*, \text{ for any } D \text{ diagonal matrix,}\end{aligned} \tag{3}$$

moreover,

$$t\left(\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2}\end{bmatrix}\right) = \begin{bmatrix} t(A_{1,1}) & A_{2,1}^* \\ A_{2,1} & t(A_{2,2})\end{bmatrix}, \tag{4}$$

where $A_{1,1}$ and $A_{2,2}$ are square matrices. We also have

$$t(A) = A - \operatorname{diag}(a_{1,1}, \dots, a_{n,n}), \qquad \text{for any } A \text{ such that } A = A^*. \tag{5}$$

We analyze some properties of the residual matrix $R = t(SAS^*) - St(A)S^*$, for S being a unitary upper Hessenberg matrix, which will be used to prove the main result in the next section. We start with a couple of technical lemmas.

Lemma 2.3. *Let $Z \in \mathbb{C}^{k\times k}$ and set $S = Z \oplus I_{n-k}$ where $n > k$. Then for any $A \in \mathbb{C}^{n\times n}$ it holds that*

$$t(SAS^*) - St(A)S^* = W \oplus 0_{n-k},$$

for some $W \in \mathbb{C}^{k\times k}$ where 0_{n-k} is the null matrix of size $n-k$. Similarly, for $S = I_{n-k} \oplus Z$ it holds that $t(SAS^*) - St(A)S^* = 0_{n-k} \oplus W'$, for some $W' \in \mathbb{C}^{k\times k}$. The same properties hold if I_{n-k} is replaced by a diagonal matrix D_{n-k}.

Proof. Concerning the first part, partition A as $A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2}\end{bmatrix}$ where $A_{1,1} \in \mathbb{C}^{k\times k}$, so that $SAS^* = \begin{bmatrix} ZA_{1,1}Z^* & ZA_{1,2} \\ A_{2,1}Z^* & A_{2,2}\end{bmatrix}$. In view of (4) we have

$$t(SAS^*) = \begin{bmatrix} t(ZA_{1,1}Z^*) & ZA_{2,1}^* \\ A_{2,1}Z^* & t(A_{2,2})\end{bmatrix}. \tag{6}$$

On the other hand,

$$St(A)S^* = S\begin{bmatrix} t(A_{1,1}) & A_{2,1}^* \\ A_{2,1} & t(A_{2,2})\end{bmatrix}S^* = \begin{bmatrix} Zt(A_{1,1})Z^* & ZA_{2,1}^* \\ A_{2,1}Z^* & t(A_{2,2})\end{bmatrix}. \tag{7}$$

So that, from (6) and (7) we get $t(SAS^*) - St(A)S^* = W \oplus 0_{n-k}$, with $W = t(ZA_{1,1}Z^*) - Zt(A_{1,1})Z^*$. The second part can be proved similarly. Finally, if I_{n-k} is replaced by the diagonal matrix D_{n-k} the same properties hold since for a diagonal matrix D one has $t(DAD^*) - Dt(A)D^* = 0$ in view of (3). □

Lemma 2.4. *Let $S = (Z \oplus I_{n-2})(1 \oplus \widehat{S})$ where $Z \in \mathbb{C}^{2\times 2}$, $\widehat{S} \in \mathbb{C}^{(n-1)\times(n-1)}$. Then for any matrix A partitioned as $A = \begin{bmatrix} a_{1,1} & u^* \\ v & \widehat{A}\end{bmatrix} \in \mathbb{C}^{n\times n}$, where $\widehat{A} \in \mathbb{C}^{(n-1)\times(n-1)}$ it holds that*

$$t(SAS^*) - St(A)S^* = W \oplus 0_{n-2} + (Z \oplus I_{n-2})(0 \oplus (t(\widehat{S}\widehat{A}\widehat{S}^*) - \widehat{S}t(\widehat{A})\widehat{S}^*))(Z^* \oplus I_{n-2}).$$

for some $W \in \mathbb{C}^{2\times 2}$.

Proof. Set $B = (1 \oplus \widehat{S})A(1 \oplus \widehat{S}^*)$ then by Lemma 2.3

$$t(SAS^*) = t((Z \oplus I_{n-2})B(Z^* \oplus I_{n-2}) = W \oplus 0_{n-2} + (Z \oplus I_{n-2})t(B)(Z^* \oplus I_{n-2}).$$

On the other hand

$$St(A)S^* = (Z \oplus I_{n-2})(1 \oplus \widehat{S})t(A)(1 \oplus \widehat{S}^*)(Z^* \oplus I_{n-2}).$$

Thus

$$t(SAS^*) - St(A)S^* = W \oplus 0_{n-2} + (Z \oplus I_{n-2})E(Z^* \oplus I_{n-2}),$$

where $E = t(B) - (1 \oplus \widehat{S})t(A)(1 \oplus \widehat{S})$. Now, since $B = (1 \oplus \widehat{S})A(1 \oplus \widehat{S}^*)$, in view of (4) we have

$$B = \begin{bmatrix} a_{1,1} & u^*\widehat{S}^* \\ \widehat{S}v & \widehat{S}\widehat{A}\widehat{S}^* \end{bmatrix}, \quad t(B) = \begin{bmatrix} 0 & v^*\widehat{S}^* \\ \widehat{S}v & t(\widehat{S}\widehat{A}\widehat{S}^*) \end{bmatrix}.$$

A similar analysis shows that

$$(1 \oplus \widehat{S})t(A)(1 \oplus \widehat{S}^*) = \begin{bmatrix} 0 & v^*\widehat{S}^* \\ \widehat{S}v & \widehat{S}t(\widehat{A})\widehat{S}^* \end{bmatrix}.$$

Thus we get

$$t(SAS^*) - St(A)S^* = W \oplus 0_{n-2} + (Z \oplus I_{n-2})(0 \oplus (t(\widehat{S}\widehat{A}\widehat{S}^*) - St(\widehat{A})\widehat{S}^*))(Z^* \oplus I_{n-2}).$$

□

A consequence of the above results is expressed by the following

Theorem 2.5. *Let $A \in \mathbb{C}^{n\times n}$, and set $Q = G_h \cdots G_k$ for $1 \le h < k \le n-1$, where the parameters s_i, c_i defining G_i are such that $s_i \neq 0$, $i = h, \dots, k$. Then $R_n := t(QAQ^*) - Qt(A)Q^* = \operatorname{diag}(d) + t(ab^*) \in \mathcal{QSH}_1^n$ for vectors $a, b, d \in \mathbb{C}^n$, where b is independent of A. More precisely, $b_h = s_h \cdots s_k$, $b_i = c_{i-1}s_i \cdots s_k$, for $i = h+1, \dots, k$, $b_i = a_i = d_i = 0$ for $i < h$ and $i > k+1$. In particular, if $h > 1$ then $R_n e_1 = 0$.*

Proof. Clearly the matrix Q has the form $I_{h-1} \oplus Z_{k-h+2} \oplus I_{n-k-1}$ where Z_{k-h+2} is a unitary Hessenberg matrix of size $k-h+2$. In view of Lemma 2.3, we can write $R_n = 0_{h-1} \oplus R_{k-h+2} \oplus 0_{n-k-1}$ and this immediately proves the last statement of the Theorem. Moreover, it follows that $a_i = b_i = d_i = 0$ for $i = 1, \dots, h-1$ and for $i = k+2, \dots, n$ so that it is sufficient to prove the claim for R_{k-h+2}. Equivalently, we may assume that $h = 1$ and $k = n-1$ so that $s_i \neq 0$ for $i = 1, \dots, n-1$. We prove that $R_n \in \mathcal{QSH}_1^n$ by induction on n. For $n = 2$ it holds that $R_2 = \left[\begin{smallmatrix} 0 & \overline{\alpha} \\ \alpha & 0 \end{smallmatrix}\right]$. This way one can choose $a_2 = \alpha/s_1$ and $b_1 = s_1$. For the inductive step, let $n > 1$ and observe that Q can be factorized as $Q = (Z \oplus I_{n-2})(1 \oplus \widehat{S})$ for $Z = \left[\begin{smallmatrix} c & s \\ -\overline{s} & c \end{smallmatrix}\right]$, and $\widehat{S} \in \mathbb{C}^{(n-1)\times(n-1)}$, where for notational simplicity we set $s = s_1$, $c = c_1$. Applying Lemma 2.4 yields

$$R_n = W \oplus 0_{n-2} + (Z \oplus I_{n-2})(0 \oplus R_{n-1})(Z^* \oplus I_{n-2})$$

for

$$W = t(Z \left[\begin{smallmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{smallmatrix}\right] Z^*) - Zt(\left[\begin{smallmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{smallmatrix}\right])Z^* = \left[\begin{smallmatrix} -c(sa_{2,1} + \overline{sa_{2,1}}) & -s(\overline{ca}_{1,1} + s\overline{a}_{1,2} - \overline{ca}_{2,2} - s\overline{a}_{2,1}) \\ -\overline{s}(ca_{1,1} + \overline{s}a_{1,2} - ca_{2,2} - \overline{s}a_{2,1}) & c(sa_{2,1} + \overline{(sa_{2,1})}) \end{smallmatrix}\right],$$

where $R_{n-1} = t(\widehat{S}\widehat{A}\widehat{S}^*) - \widehat{S}t(\widehat{A})\widehat{S}^*$ and $\widehat{A}$ is the trailing principal submatrix of A of size $n-1$. A direct inspection shows that

$$R_n = W \oplus 0_{n-2} + \left[\begin{array}{c|c} |s|^2 e_1^T R_{n-1} e_1 & s e_1^T R_{n-1} D \\ \hline \overline{s} D R_{n-1} e_1 & D R_{n-1} D \end{array}\right], \quad D = c \oplus I_{n-2}. \tag{8}$$

From the inductive assumption we may write that $R_{n-1} = \mathrm{diag}(\widehat{d}) + t(\widehat{a}\widehat{b}^*)$ for $\widehat{a}, \widehat{b}, \widehat{d} \in \mathbb{C}^{n-1}$, where $\widehat{b}_1 = s_2 \cdots s_{n-1} \neq 0$, $\widehat{b}_i = c_i s_{i+1} \cdots s_{n-1}$. So that (8) turns into

$$R_n = \left[\begin{array}{cccc} w_{1,1} & w_{1,2} & & \\ w_{2,1} & w_{2,2} & & \\ & & & \\ & & & \end{array}\right] + \left[\begin{array}{c|c|cccc} |s|^2\widehat{d}_1 & * & * & \cdots & \cdots & * \\ \hline \widehat{d}_1 c\overline{s} & c^2\widehat{d}_1 & * & \cdots & \cdots & * \\ \hline \widehat{a}_2\overline{\widehat{b}}_1\overline{s} & \widehat{a}_2\overline{\widehat{b}}_1 c & \widehat{d}_2 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \widehat{a}_3\overline{\widehat{b}}_2 & \widehat{d}_3 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & * \\ \widehat{a}_{n-1}\overline{\widehat{b}}_1\overline{s} & \widehat{a}_{n-1}\overline{\widehat{b}}_1 c & \widehat{a}_{n-1}\overline{\widehat{b}}_2 & \cdots & \widehat{a}_{n-1}\overline{\widehat{b}}_{n-2} & \widehat{d}_{n-1} \end{array}\right],$$

where the upper triangular part, denoted with $*$, is determined by symmetry. Thus, it follows that $R_n = \mathrm{diag}(d) + t(ab^*)$ where $d_1 = |s_1|^2\widehat{d}_1 + w_{1,1}$, $d_2 = c_1^2\widehat{d}_1 + w_{2,2}$, $d_i = \widehat{d}_{i-1}$, for $i = 3, \ldots, n$; moreover

$$\begin{aligned} a_2 &= \frac{1}{\overline{\widehat{b}}_1}(c\overline{s}\widehat{d}_1 + w_{2,1}), \\ a_i &= \widehat{a}_{i-1}, \ \text{ for } i = 3, \ldots, n, \\ b_1 &= s_1\widehat{b}_1, \ \ b_2 = c_1\widehat{b}_1, \\ b_i &= \widehat{b}_{i-1}, \ \text{ for } i = 3, \ldots, n-1, \end{aligned}$$

where the condition $s_i \neq 0$ implies that $b_1 \neq 0$. This completes the proof. □

The property that $R = D + t(ab^*)$, where b is independent of A enables us to prove the following

Corollary 2.6. *In the hypotheses of Theorem 2.5 we have*

$$R = t(QAQ^*) - Q(D + t(A))Q^* \in \mathcal{QSH}_1^n$$

for any diagonal matrix D with real entries. Moreover, $R = \mathrm{diag}(d) + t(ab^*)$ for some vectors $a, b \in \mathbb{C}^n$ and $d \in \mathbb{R}^n$.

Proof. Without loss of generality we may assume that $s_i \neq 0$ for $i = 1, \ldots, n-1$. In view of Theorem 2.5, $t(QAQ^*) - Qt(A)Q^* = \mathrm{diag}(d) + t(ab^*)$, where b is independent of A. On the other hand equation (5) implies that $QDQ^* = \mathrm{diag}(d') + t(QDQ^*)$, moreover, since $t(D) = 0$, Theorem 2.5 implies that $t(QDQ^*) = t(a'b^*)$ for some vector a'. Thus we get $QDQ^* = \mathrm{diag}(d') + t(a'b^*)$. Therefore $R = \mathrm{diag}(d) + t(ab^*) - \mathrm{diag}(d') - t(a'b^*) = \mathrm{diag}(d + d') + t((a - a')b^*)$, that is, $R \in \mathcal{QSH}_1^n$. □

A further analysis enables us to provide an explicit representation of the ith row of the matrix R. More precisely we have the following result:

Theorem 2.7. *Under the assumptions of Theorem 2.5, the ith row of the matrix $R = t(QAQ^*) - Q(D + t(A))Q^*$ has the representation*

$$e_i^T R = [0, \ldots, 0, v_i, d_i, w_i^*] G_{i-2}^* G_{i-3}^* \cdots G_1^* \tag{9}$$

where, $v_i, d_i \in \mathbb{C}$, $w_i \in \mathbb{C}^{n-i}$ and $d_i = r_{i,i}$.

Proof. Let us write $Q = Q_1 Q_2$ where $Q_1 = G_1 \cdots G_{i-2}$, $Q_2 = G_{i-1} \cdots G_{n-1}$, so that (9) can be rewritten as $e_i^T R Q_1 = [0, \ldots, 0, v_i, d_i, w_i^*]$. This way, it is enough to show that the ith row of RQ_1 has the first $i-2$ entries equal to zero. In view of Lemma 2.3 we have

$$R' := t(Q_2 A Q_2^*) - Q_2(D + t(A))Q_2^* = 0_{i-2} \oplus \widehat{R} \in \mathcal{QSH}_1^n.$$

Whence

$$Q_2(D + t(A))Q_2^* = t(Q_2 A Q_2^*) - 0_{i-2} \oplus \widehat{R}. \tag{10}$$

Moreover, by definition of R we have

$$RQ_1 = t(Q_1 Q_2 A Q_2^* Q_1^*)Q_1 - Q_1 Q_2 (D + t(A))Q_2^*.$$

Setting $B = Q_2 A Q_2^*$ and combining the above equation with (10) yields

$$RQ_1 = t(Q_1 B Q_1^*)Q_1 - Q_1 t(B) + Q_1(0_{i-2} \oplus \widehat{R}).$$

Now, since $Q_1(0_{i-2} \oplus \widehat{R})$ has the first $i-2$ columns equal to zero, it is sufficient to prove that the ith row of $t(Q_1 B Q_1^*)Q_1 - Q_1 t(B)$ has the first $i-2$ components zero. To this regard, observe that $Q_1 = \widetilde{Q}_1 \oplus I_{n-i+1}$, where $\widetilde{Q}_1 \in \mathbb{C}^{(i-1)\times(i-1)}$, so that partitioning B as $\begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}$, where $B_{1,1} \in \mathbb{C}^{(i-1)\times(i-1)}$, by applying again Lemma 2.3 we find that $t(Q_1 B Q_1^*) - Q_1 t(B) Q_1^* = W_{i-1} \oplus 0_{n-i+1}$ for some $W_{i-1} \in \mathbb{C}^{(i-1)\times(i-1)}$. This implies that $t(Q_1 B Q_1^*)Q_1 - Q_1 t(B)$ has the last $(n-i+1)$ rows equal to zero. This completes the proof. □

Observe that the representation of R that we have found is exactly the Givens-Vector representation for quasiseparable matrices that is presented in [13]. A more general analysis of this representation is given in Section 4.

3. Reduction to Hessenberg form

It is a simple matter to show that the Hessenberg form $H = QAQ^*$ maintains a quasiseparable structure. In fact, since D is real, we find that $H = QDQ^* + QUV^*Q^*$ is the sum of a Hermitian matrix $S = QDQ^*$ and of a matrix $T = QUV^*Q^*$ of rank k. Since H is Hessenberg, the submatrices contained in the strictly lower triangular part of H have rank at most 1 so that the submatrices in the strictly lower triangular part of $S = H - T$ have rank at most $k+1$. On the other hand, since S is Hermitian, then also the submatrices contained in its strictly upper triangular part have rank at most $k+1$, thus the submatrices in the strictly upper triangular part of $H = S+T$ have rank at most $2k+1$ being the sum of the ranks of the corresponding submatrices of S and T respectively. This way we find that H is $(1, 2k+1)$-quasiseparable. Actually, we will see in Section 3.1 that the submatrices contained in the strictly upper triangular part of H have rank $2k-1$.

The nontrivial problem is to exploit this structure property and to provide a way to compute the matrix H with a cost lower than $O(n^3)$ ops needed for general matrices. To this end we have to recall the customary procedure for reducing a matrix A into Hessenberg form which is

Algorithm 1 Reduction to Hessenberg form by means of Givens rotations

```
for $j = 1, \dots, n-2$ do
    for $i = n, \dots, j+2$ do
        $G \leftarrow \texttt{givens}(A[i-1, j], A[i, j])$
        $A[i-1:i, :] \leftarrow G \cdot A[i-1:i, :]$
        $A[:, i-1:i] \leftarrow A[:, i-1:i] \cdot G^*$
    end for
end for
```

based on Givens rotations. The algorithm can be easily described by the pseudo-code reported in Algorithm 1 where the function $\texttt{givens}(v_1, v_2)$ provides the matrix $G(v_1, v_2)$.

Denote by $G_{i-1,j}$ the unitary $n \times n$ matrix which performs the Givens rotation $G = \texttt{givens}(A(i-1, j), A(i, j))$ in the rows $i-1$ and i at step j of Algorithm 1 and set $Q_j = G_{j+1,j} G_{j+2,j} \cdots G_{n-1,j}$. Then Algorithm 1 generates a sequence A_j of matrices such that

$$\begin{aligned} &A_0 = A, \quad A_{n-2} = H \\ &A_j = Q_j A_{j-1} Q_j^*, \quad j = 1, \dots, n-2 \end{aligned}$$

where the matrix A_j has the form

$$A_j = Q_j \dots Q_1 (D + UV^*) Q_1^* \dots Q_j^* = \widehat{D} + \widehat{U}\widehat{V}^* = \begin{bmatrix} a_{1,1}^{(j)} & \dots & a_{1,j}^{(j)} & \times & \dots & \times \\ a_{2,1}^{(j)} & \ddots & & \vdots & & \vdots \\ & \ddots & a_{j,j}^{(j)} & \times & \dots & \times \\ & & a_{j+1,j}^{(j)} & \star & \dots & \star \\ & & & \vdots & & \vdots \\ & & & \star & \dots & \star \end{bmatrix}, \tag{11}$$

and the symbols × and $\star$ denote arbitrary numbers.

For notational simplicity, we denote by $\widehat{A}_j$ the $(n-j) \times (n-j)$ trailing principal submatrix of A_j, that is, the submatrix represented by $\star$ in (11). Observe that $\widehat{A}_j$ is the part of A_j that has not yet been reduced to Hessenberg form. Finally, by following the notation of Section 2, we write G_i or $G_{i,j}$ to denote a unitary matrix which applies a Givens transformation in the rows i and $i+1$.

The following lemma is useful to get rid of non-generic cases in the process of Hessenberg reduction.

Lemma 3.1. *Let $v \in \mathbb{C}^n$, $v \neq 0$ and consider Givens rotations $G_1, \dots, G_{n-1}$ constructed in such a way that $(G_i \dots G_{n-1})v = (w^{(i)*}, 0, \dots, 0)^*$, where $w^{(i)} \in \mathbb{C}^i$, for $i = 1, \dots, n-1$. If there exists h such that $G_h = I$ then one can choose $G_i = I$ for every $i \geqslant h$, that is, $(G_1 \cdots G_{h-1})v = (w_1^{(1)*}, 0, \dots, 0)^*$.*

Proof. Since $G_i \cdots G_{n-1}$ is a unitary matrix which acts in the last $n-i+1$ components of v, the 2-norm of $v[i:n]$ coincides with the 2-norm of $(G_i \cdots G_{n-1} v)[i:n]$, that is, $|w_i^{(i)}|$. On the other hand, if $G_h = I$ then $(w^{(h)*}, 0, \dots, 0) = (w^{(h+1)*}, 0, \dots, 0)$ so that $w_{h+1}^{(h+1)} = 0$. This implies that $\|v[h+1:n]\| = 0$, whence $v_i = 0$ for $i = h+1, \dots, n$. This way, one can choose $G_i = I$ for $i = h+1, \dots, n-1$. □

Observe that in view of Lemma 2.2 the unitary matrix $Q = G_2 \cdots G_{n-1}$ is in upper Hessenberg form. Moreover, the rotations G_i are such that $QAe_1 = \alpha e_1 + \beta e_2$ for some α, β. In view of Lemma 3.1 we can assume that if $G_h = I$ then $G_i = I$ for every $i \geqslant h$, that is, $Q = G_2 \ldots G_{h-1}$.

The quasiseparable structure of the matrices $\widehat{A}_i$ is a consequence of our main result which is reported in the following

Theorem 3.2. *Let $U, V, W \in \mathbb{C}^{n\times k}$, $S = \operatorname{diag}(d) + t(ab^*) \in \mathcal{QSH}_1^n$ and define*

$$A = UV^* + t(UW^*) + S.$$

Let G_i, $i = 2, \ldots, n-1$ be Givens rotations acting on the rows i and $i+1$ such that

$$QAe_1 = a_{1,1}e_1 + \beta e_2, \quad \text{where } Q = G_2 \ldots G_{n-1}.$$

Then the matrix $\widehat{A}$ obtained by removing the first row and the first column of QAQ^* can be written again as

$$\widehat{A} = \widehat{U}\widehat{V}^* + t(\widehat{U}\widehat{W}^*) + \widehat{S}$$

where $\widehat{U}, \widehat{W} \in \mathbb{C}^{(n-1)\times k}$, and $\widehat{S} = \operatorname{diag}(\widehat{d} + t(\widehat{a}\widehat{b}^*)) \in \mathcal{QSH}_1^{n-1}$ for some vectors $\widehat{d}, \widehat{a}, \widehat{b} \in \mathbb{C}^{n-1}$. Moreover, $\widehat{U}$ and $\widehat{V}$ are obtained by removing the first row of QU and QV, respectively.

Proof. According to Lemma 3.1, we may assume that in the first step of the process of reduction in Hessenberg form, the parameters s_i satisfy the condition $s_i \neq 0$ for $i = 2, \ldots, h$, while $s_i = 0$, for $i = h+1, \ldots, n-1$, for some $h \leq n-1$. Note that in this case $Q = \widetilde{Q} \oplus I_{n-h-1}$ so we can apply this theorem to the matrix $\widetilde{Q}\widetilde{A}\widetilde{Q}^*$ where $\widetilde{A}$ is the leading square block of A partitioned according to the partitioning of Q. Lemma 2.3 provides a way to extend this representation to dimension n. In view of this fact we can assume, without loss of generality, that $h = n-1$. We have

$$QAQ^* = (QU)(QV)^* + F, \qquad F = Q(t(UW^*) + S)Q^*.$$

In view of Corollary 2.6 we have $F = t(Q(UW^* + ab^*)Q^*) - R$ for $R \in \mathcal{QSH}_1^n$. Thus

$$QAQ^* = (QU)(QV)^* + t(QU(QW)^*) + t(Qa(Qb)^*) - R. \tag{12}$$

Recall from Theorem 2.5 that $Re_1 = 0$ and that Q has been chosen so that $QAQ^*e_1 = \alpha e_1 + \beta e_2$. This fact, together with (12), implies that the vector $u = t(Qa(Qb)^*)e_1$ is such that $u[3:n]$ is in the span of the columns of $(QU)[3:n,:]$. In view of (4) we may write $t(Qa(Qb)^*)[2:n, 2:n] = t(\widehat{u}z^*)$ for $\widehat{u} = u[2:n]$, and for a suitable $z \in \mathbb{C}^{n-1}$. Applying (3) yields the following representation for the trailing principal submatrix $\widehat{A}$ of QAQ^* of size $n-1$

$$\widehat{A} = \widehat{U}\widehat{V}^* + t(\widehat{U}\widetilde{W}^* + \widehat{u}\widehat{z}^*) - \widehat{R}$$

where $\widehat{U}$, $\widehat{V}$ and $\widetilde{W}$ are obtained by removing the first row of U, V and W, respectively, while $\widehat{R} = R[2:n, 2:n]$. Since $\widehat{u}[2:n]$ is in the span of $\widehat{U}[2:n,:]$, and since the first row of $\widehat{U}$ as well as the first entry of $\widehat{u}$ do not play any role in the value of $t(\widehat{U}\widetilde{W}^* + \widehat{u}\widehat{z}^*)$, we may set $\widehat{u}_1$ equal to an appropriate value in such a way that $\widehat{u}$ is in the span of the columns of $\widehat{U}$. This way, the matrix $\widehat{U}\widetilde{W}^* + \widehat{u}\widehat{z}^*$ has rank at most k and can be written as $\widehat{U}\widehat{W}^*$ for a suitable $\widehat{W} \in \mathbb{C}^{k\times n}$. Thus we have

$$\widehat{A} = \widehat{U}\widehat{V}^* + t(\widehat{U}\widehat{W}^*) + \widehat{S}$$

for $\widehat{S} = -\widehat{R}$, that concludes the proof. □

Observe that the matrix A defined in (1) satisfies the assumptions of the above theorem with $W = 0$ and $S = D$ real diagonal. This way, Theorem 3.2 shows that the trailing principal submatrix $\widehat{A}_j$ of the sequence generated by the Hessenberg reduction, maintains the structure $\widehat{A}_j = U_j V_j^* + t(U_j W_j^*) + S_j$ where $S_j = \mathrm{diag}(d_j) + t(a_j b_j^*)$.

This fact is fundamental to design fast algorithms for the Hessenberg reduction of a matrix of the form (1). In order to realize this goal, we need to choose a reliable explicit representation for $t(U_j W_j^*)$ and S_j. This is the topic of the next section.

Note also that the representation $\widehat{A}_j = U_j V_j^* + t(U_j W_j^*) + S_j$ implies that $\widehat{A}_j$ is still a quasiseparable matrix of low quasiseparability rank. In fact, $t(U_j W_j^*) + U_j V_j^*$ is (at most) $(k, 2k)$-quasiseparable, where k is the number of columns of U_j. The sum with S_j provides a $(k+1, 2k+1)$-quasiseparable matrix. We will prove in the next Theorem 3.3 that after the first step it is possible to replace U_j, V_j and W_j with $n \times (k-1)$ matrices, thus showing that $\widehat{A}_j$ is a $(k-1, 2k-1)$-quasiseparable matrix at all the intermediate steps of the Hessenberg reduction.

3.1. Analysis of the first step

Recall that the matrix $A = A_0$ is of the kind $A = D + UV^*$. This is a particular case of the form $A = UV^* + t(UW^*) + S$ above where $W = 0$ and S is diagonal. Because of this additional structure, the first Hessenberg reduction step is somewhat special and we can get a sharper bound for the quasiseparability rank of $\widehat{A_1}$. This is shown in the next result.

Theorem 3.3. *Assume that the matrix A of* (1) *does not have the first column already in Hessenberg form, i.e., $A[3:n,1] \neq 0$. Then the rank of the matrix obtained by removing the first two rows of $Q_1 U$ is less than $k-1$. Moreover, the $(n-1) \times (n-1)$ trailing principal submatrix $\widehat{A}_1$ of $A_1 = Q_1 A Q_1^*$ has lower quasiseparable rank k.*

Proof. Observe that $\widehat{A}_1$ can be written as the sum of a Hermitian matrix and of a matrix of rank at most k, namely, $\widehat{A}_1 = \widehat{Q}_1 D \widehat{Q}_1^* + \widehat{U}_1 \widehat{V}_1^*$, where $\widehat{U}_1$, $\widehat{V}_1$ and $\widehat{Q}_1$ are the matrices obtained by removing the first row of $Q_1 U$, $Q_1 V$, and Q_1, respectively. Define $x = V^* e_1$ so that $x \neq 0$ and $Ae_1 = d_1 e_1 + Ux$. Since $Q_1 A e_1 = \alpha_1 e_1 + \beta_1 e_2$, we find that $Q_1 U x = Q_1 A e_1 - d_1 Q_1 e_1 = (\alpha_1 - d_1) e_1 + \beta_1 e_2$, which implies $\widehat{U}_1 x = \beta_1 e_1$. Thus, $\widehat{U}_1[2:n-1,:]$ has rank at most $k-1$. This matrix coincides with the matrix obtained by removing the first two rows of $Q_1 U$ so that the first part of the theorem is proven. Since $\widehat{U}_1[2:n-1,:]$ has rank at most $k-1$, then also the matrix $\widehat{U}_1[2:n-1,:]\widehat{V}_1^*$ has rank at most $k-1$. We can conclude that every submatrix in the strictly lower triangular part of $\widehat{A}_1$, given by the sum of a submatrix of $\widehat{U}_1[2:n-1,:]\widehat{V}_1^*$ which has rank at most $k-1$, and a submatrix in the lower triangular part of $Q_1 D Q_1^*$ which has rank at most 1 in view of Corollary 2.6, can have rank at most k. This completes the proof. □

It is important to note that we are not tracking the structure of the whole matrix but only the structure of the trailing part that we still need to reduce. This is not a drawback since the trailing part is the only information needed to continue the algorithm. Moreover, the entire Hessenberg matrix can be recovered at the end of the process by just knowing the diagonal and subdiagonal elements that are computed at each step together with the matrices U_{n-2} and V_{n-2}.

4. Representing quasiseparable matrices

Finding good and efficient representations of quasiseparable matrices is a problem that has been studied in recent years. Some representations have been introduced and analyzed for the rank 1 case. Some of them have been extended to (or were originally designed for) the higher rank case [5, 6, 12, 13].

In this section we provide a representation of quasiseparable matrices which, combined with the results of the previous section, enables us to design an algorithm for the Hessenberg reduction of the matrix A with cost $O(n^2k)$.

4.1. Givens Vector representations

A useful family of representations for quasiseparable matrices is described in [13] (for the 1-quasiseparable case) and is extended to a more general version in [10]. We use this kind of representation and adjust it to our framework. For the sake of clarity, we provide the details of the new notation in order to make this section self-contained.

Definition 4.1. A tuple $\mathcal{G} = (G_i)_{i \in I}$ on some ordered index set $(I, \leqslant)$ is said a *sequence* of Givens rotations. We also define $\prod_{i \in I} G_i$ the product in increasing order while $\prod_{i \in \mathrm{rev(I)}} G_i$ denotes the product in decreasing order with respect to the order defined on I. The following operations on $\mathcal{G}$ are introduced:

- $\mathcal{G}v := \prod_{i \in \mathrm{rev(I)}} G_i v$, for $v \in \mathbb{C}^n$;
- $\mathcal{G}^* v := \prod_{i \in I} G_i^* v$ for $v \in \mathbb{C}^n$;
- for $J \subseteq I$, with the induced order, we call $\mathcal{G}[J] := (G_j)_{j \in J}$ the *slice* of $\mathcal{G}$ on the indices J;
- for Givens sequences $\mathcal{G} = (G_i)_{i \in I}$, $\mathcal{G}' = \left(G'_j\right)_{j \in J}$, we define the product $\mathcal{G}\mathcal{G}'$ to be the sequence
$$\mathcal{G}\mathcal{G}' := (E_i)_{i \in I \sqcup J}\,, \qquad E_i = \begin{cases} G_i & \text{if } i \in I \\ G'_i & \text{if } i \in J \end{cases}$$
where $\sqcup$ is the disjoint union operator and where the order on $I \sqcup J$ is induced by the ones on I and J and by the agreement that $G_i < G'_j$ for every $i \in I, j \in J$.

The above definitions on the product between a sequence and a vector trivially extend to products between sequences and matrices. For instance, $\mathcal{G}A := \prod_{i \in \mathrm{rev(I)}} G_i A$.

We are interested in the cases where the index sets I are special, in particular we consider the case of univariate sets $I \subset \mathbb{N}$, and the case of bivariate sets $I \subset \mathbb{N}^2$. Let us give first the more simple definition of 1-sequences, which covers the case of univariate sets, and then extend it to k-sequences for $k > 1$, that is, the case of bivariate sets.

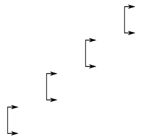
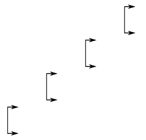
Definition 4.2. We say that $\mathcal{G}$ is a 1-*sequence* of Givens rotations if $\mathcal{G} = (G_2, \ldots, G_{n-1})$.

Notice that in this context, the operations already introduced in Definition 4.1 specialize in the following way:

- $\mathcal{G}v := G_{n-1} \ldots G_2 v$, for $v \in \mathbb{C}^n$;
- $\mathcal{G}^* v := G_2^* \ldots G_{n-1}^* v$, for $v \in \mathbb{C}^n$;
- $\mathcal{G}[i:j] := (G_i, \ldots, G_j)$, for $2 \leqslant i < j \leqslant n-1$, is a *slice* of $\mathcal{G}$ from i to j.

Here, and hereafter, we use the notation $i : j$ to mean the tuple $(i, i+1, \ldots, j)$ for $i < j$. Sometimes we use the expressions $\mathcal{G}[: j]$ and $\mathcal{G}[i :]$ for $\mathcal{G}[2 : j]$ and $\mathcal{G}[i : n-1]$, respectively. That is, leaving the empty field before and after the symbol ":" is a shortcut for "starting from the first rotation" and for "until the last rotation", respectively.

Below, we recall a useful pictorial representation, introduced in [13] and [10], which effectively describes the action of the sequence of Givens rotations. We report the case $n = 6$, where every describes a Givens rotation G_i applied to the pair $(i, i+1)$ of consecutive rows.

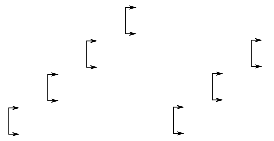
$\mathcal{G} = (G_2, \dots, G_n) =$  , $\mathcal{G}v =$  $\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix}.$

The definition of 1-sequences is generalized to the case where $I \subset \mathbb{N}^2$ is a bivariate set of indices, and where we consider Givens rotations $G_{i,j}$ acting on the pair $(i, i+1)$ of consecutive rows for any j. In this case, the ordering on $\mathbb{N}^2$ which induces orderings in any subset I of $\mathbb{N}^2$, is defined by

$$(i_1, j_1) \leqslant_G (i_2, j_2) \iff j_1 > j_2 \text{ or } (j_1 = j_2 \text{ and } i_1 \leqslant i_2) .$$

Definition 4.3. We say that $\mathcal{G} = (G_{i,j})_{(i,j)\in I}$ is a k-sequence of Givens rotations if $I = \{(i, j) \in \mathbb{N}^2 \mid i = 2, \dots, n-1, j = 1, \dots, \min(i-1, k)\}$ with the order induced by $\leqslant_G$. With a slight abuse of notation we define the sequence $\mathcal{G}[i_1 : i_2] := (G_{i,j})_{(i,j)\in I'}$, $I' = \{(i, j) \in \mathbb{N}^2 \mid i = i_1, \dots, \min(i_2 + k, n-1), j = \max(1, i - i_2 + 1), \dots, \min(k, i - i_1 + 1), \quad 2 \leqslant i_1 < i_2 \leqslant n-1\}$ to be a *slice* of $\mathcal{G}$ from i_1 to i_2, where the ordering in I' is induced by the ordering $\leqslant_G$ valid on the parent set.

A pictorial representation similar to the one given above can be used also in this case. For example, for $k = 2$ and $n = 6$ we have $\mathcal{G} = (G_{3,2}, G_{4,2}, G_{5,2}, G_{2,1}, G_{3,1}, G_{4,1}, G_{5,1})$ that is represented by

$\mathcal{G} =$  . (13)

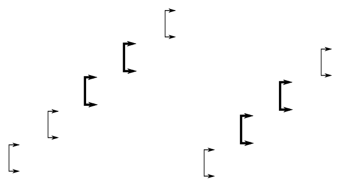
Note that for every $i_1 \leqslant i_2 < i_3$, the slices of $\mathcal{G}$ can be factored in the following form:

$$\mathcal{G}[i_1 : i_3] = \mathcal{G}[i_2 + 1 : i_3]\mathcal{G}[i_1 : i_2], \qquad \mathcal{G}^*[i_1 : i_3] = \mathcal{G}^*[i_1 : i_2]\mathcal{G}^*[i_2 + 1 : i_3],$$

where, for notational simplicity, we set $\mathcal{G}^*[i_1 : i_2] = (\mathcal{G}[i_1 : i_2])^*$. This property is called *slicing of rotations.*

Note that the order $\leqslant_G$ is one of the orders such that $\mathcal{G}v$ coincides with the multiplication of the vector v by the Givens rotations in $\mathcal{G}$ with the order induced by the pictorial representation (13).

It is worth highlighting that the operation of slicing a k-sequence is equivalent to removing the heads and tails from the sequences itself. For example the slice of $\mathcal{G}$ defined by $\mathcal{G}[3 : n-2]$ is obtained by taking only the bold rotations in the following picture, where $n = 7$, which correspond to $G_{4,2}, G_{5,2}, G_{3,1}, G_{4,1}$.

$\mathcal{G} =$  .

With the basic tools introduced so far we can define the concept of Givens Vector representation.

Definition 4.4. A *Givens Vector (GV) representation* of rank k for a Hermitian quasiseparable matrix A is a triple $(\mathcal{G}, W, D)$ where $\mathcal{G}$ is a k-sequence of Givens rotations, $W \in \mathbb{C}^{k\times(n-1)}$ and D is a diagonal matrix such that

- D is the diagonal of A;
- for every $i = 1, \dots, n-1$ the subdiagonal elements of the i-th column of A are equal to the last $n-i$ elements of $\mathcal{G}[i+1:]\underline{w}_i$, where we define

$$\underline{w}_i := \begin{bmatrix} 0_i \\ We_i \\ 0_{n-k-i} \end{bmatrix} \text{ if } i < n-k, \qquad \underline{w}_i := \begin{bmatrix} 0_i \\ (We_i)[1:n-i] \end{bmatrix} \text{ otherwise}$$

  where 0_j is the 0 vector of length j if $j > 0$, and is the empty vector otherwise. That is, $\operatorname{tril}(A, -1)e_i = \mathcal{G}[i+1:]\underline{w}_i$.

If the triple $(\mathcal{G}, W, D)$ is a GV representation of the matrix A we write $A = \mathrm{GV}(\mathcal{G}, W, D)$.

We refer to [10] for a detailed analysis of the properties of this representation. We recall here only the following facts:

- If A is k-quasiseparable then there exists a k-sequence $\mathcal{G}$, a matrix $W \in \mathbb{C}^{k\times(n-1)}$ and a diagonal matrix D such that $A = \mathrm{GV}(\mathcal{G}, W, D)$.
- If $A = \mathrm{GV}(\mathcal{G}, W, D)$ for some k-sequence $\mathcal{G}$, $W \in \mathbb{C}^{k\times(n-1)}$ and D diagonal, then A is at most k-quasiseparable.

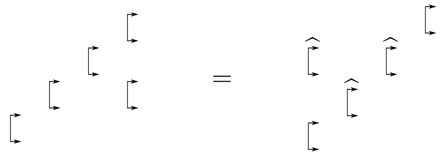
We introduce now an important operation on Givens rotations, called *turnover*. The following Lemma can be thought as a partial answer to the question whether two Givens rotations commute. It is clear that if we have G_i and G_j such that $|i-j| > 1$ then $G_iG_j - G_jG_i = 0$. This is also true if the two rotations act on the same rows, but it does not hold when they are acting on consecutive rows. In the latter case, the turnover gives a way to swap the order of the rotations.

Lemma 4.5. *Let $\mathcal{G}$ be a sequence of Givens rotations and F_i a Givens rotation acting on the rows i and $i+1$. Then there exists another sequence $\widehat{\mathcal{G}}$ and a Givens rotation $\widehat{F}_{i-1}$ acting on the rows $i-1$ and i such that*

$$\mathcal{G}F_i = \widehat{F}_{i-1}\widehat{\mathcal{G}}.$$

Moreover, $\widehat{\mathcal{G}}$ differs from $\mathcal{G}$ only in the rotations acting on the rows with indices $(i-1, i)$ and $(i, i+1)$.

See [13] for a proof of this fact. The pictorial representation of the Givens rotations can be helpful to understand how the turnover works.



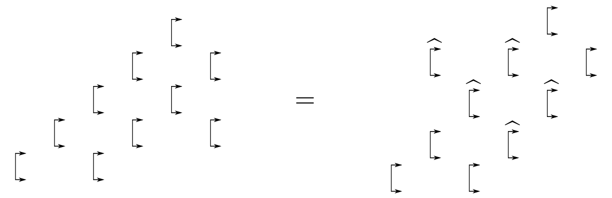
The above lemma can be easily extended to k-sequence of rotations.

Corollary 4.6. *Let $\mathcal{G}$ be a k-sequence of Givens rotations and F_i a Givens rotation acting on the rows i and $i+1$. Then there exists another k-sequence $\widehat{\mathcal{G}}$ and a Givens rotation $\widehat{F}_{i-k}$ acting on the rows $i-k$ and $i-k+1$ such that*

$$\mathcal{G}F_i = \widehat{F}_{i-k}\widehat{\mathcal{G}},$$

where $\widehat{F}_{i-k} = I$ if $i-k \leq 0$. Moreover, $\widehat{\mathcal{G}}$ differs from $\mathcal{G}$ only in the rotations of indices $(i-j+1, j)$ and $(i-j, j)$ for $j = 1, \ldots, k$.

Again, a pictorial representation of this fact can be useful to figure out the interplay of the rotations. Below, we report the case where $i > k+1$.



The above operations are very cheap. The cost of the computation of a turnover is $O(1)$ in case of 1-sequences and $O(k)$ in case of k-sequences.

We answer now to the following question: Given a k-sequence $\mathcal{G}$ and a k-quasiseparable matrix A, there exist appropriate matrices $W \in \mathbb{C}^{k\times(n-1)}$ and D diagonal such that $A = \mathrm{GV}(\mathcal{G}, W, D)$?

Lemma 4.7. *Let A be a Hermitian matrix and $\mathcal{G}$ a k-sequence of Givens rotations. Then $B = \mathcal{G}^*A$ is lower banded with a bandwidth of k, i.e., $b_{i,j} = 0$ for $i - j > k$, if and only if the matrix A admits a representation of the form $\mathrm{GV}(\mathcal{G}, W, D)$ for some $W \in \mathbb{C}^{k\times n}$ and D real diagonal.*

Proof. We first suppose that $A = \mathrm{GV}(\mathcal{G}, W, D)$. Recall that, by definition of GV representation, $\mathrm{tril}(A, -1)e_i = \mathcal{G}[i+1:]\underline{w}_i$ for $i = 1, \ldots, n-1$. This implies that

$$\mathcal{G}^*\,\mathrm{tril}(A,-1)e_i = \mathcal{G}^*\mathcal{G}[i+1:]\underline{w}_i = \mathcal{G}^*[:i]\mathcal{G}^*[i+1:]\mathcal{G}[i+1:]\underline{w}_i = \mathcal{G}^*[:i]\underline{w}_i.$$

We also have

$$\mathcal{G}^*\,\mathrm{triu}(A)e_i = \mathcal{G}^*[:i]\mathcal{G}^*[i+1:]\,\mathrm{triu}(A)e_i = \mathcal{G}^*[:i]\,\mathrm{triu}(A)e_i,$$

since $G^*[i+1:]$ is acting on rows that are null. So by decomposing $A = \mathrm{tril}(A,-1) + \mathrm{triu}(A)$ we have

$$\mathcal{G}^*Ae_i = \mathcal{G}^*\,\mathrm{triu}(A)e_i + \mathcal{G}^*\,\mathrm{tril}(A,-1)e_i = \mathcal{G}^*[:i](\mathrm{triu}(A)e_i + \underline{w}_i).$$

Now observe that the rotations inside $\mathcal{G}^*[:i]$ only act on the first $i+k$ rows. This implies that, since both $\underline{w}_i$ and $\mathrm{triu}(A)e_i$ have all the components with index strictly bigger than $i+k$ equal to zero, the same must hold for $\mathcal{G}^*[:i](\underline{w}_i + \mathrm{triu}(A)e_i)$, and this completes the proof. The converse is also true. In fact, if $\mathcal{G}^*A$ is lower banded with bandwidth k we can build W by setting $We_i = (\mathcal{G}^*[i+1:]Ae_i)\,[i+1:i+k]$ and D equal to the diagonal of A. Then the equation $A = \mathrm{GV}(\mathcal{G}, W, D)$ can be verified by direct inspection. □

To simplify the notation when talking about ranks in the lower part of quasiseparable matrices, we say that a k-sequence $\mathcal{G}$ *spans* $U \in \mathbb{C}^{n\times k}$ if there exists $Z \in \mathbb{C}^{k\times k}$ such that $\mathcal{G}^*U = \left[\begin{smallmatrix} Z \\ 0 \end{smallmatrix}\right]$. This definition is motivated by the following

Lemma 4.8. *If $\mathcal{G}$ spans $U \in \mathbb{C}^{n\times k}$ then, for every $V \in \mathbb{C}^{n\times k}$, $W \in \mathbb{C}^{k\times(n-1)}$ and D diagonal, the matrix $A_1 = UV^* + \mathrm{GV}(\mathcal{G}, W, D)$ is lower k-quasiseparable and $A_2 = t(UV^*) + \mathrm{GV}(\mathcal{G}, W, D)$ is k-quasiseparable. In particular, both $\mathcal{G}^* A_1$ and $\mathcal{G}^* A_2$ are lower banded with bandwidth k.*

Proof. For the first part of the Lemma it suffices to observe that $\mathcal{G}^* A_1$ is lower banded with bandwidth k. This follows directly by noting that $A = \mathrm{GV}(\mathcal{G}, W, D) + UV^*$. Since $\mathcal{G}^* UV^* = \left[\begin{smallmatrix} Z \\ 0 \end{smallmatrix}\right] V^*$ and $\mathcal{G}^*\mathrm{GV}(\mathcal{G}, W, D)$ is lower banded by Lemma 4.7, we conclude that also $\mathcal{G}^* A_1$ is lower banded with bandwidth k. Since the strictly lower part of A_2 coincides with the one of A_1 we find that also A_2 is lower k-quasiseparable. Given that A_2 is Hermitian, we conclude that A_2 is also upper k-quasiseparable. To see that also $\mathcal{G}^* A_2$ is lower banded we can write

$$\mathcal{G}^* A_2 = \mathcal{G}^*(A_1 - \mathrm{triu}(UV^*) + \mathrm{triu}(VU^*, 1)) = \mathcal{G}^* A_1 + \mathcal{G}^* R,$$

where R is upper triangular. Since $\mathcal{G}^*$, represented as a matrix, is the product of k upper Hessenberg matrices, it is lower banded with bandwidth k. This implies that also $\mathcal{G}^* R$ is lower banded with bandwidth k and so the same must hold also for $\mathcal{G}^* A_2$. □

Remark 4.9. *The above Lemma shows how the Givens rotations in a GV representation of a matrix in $\mathcal{QSH}_k$ are sufficient to determine the column span of the submatrices contained in the lower triangular part. These matrices give the same information obtained by knowing the matrix U in the $D + t(UV^*)$ representation.*

We need to find efficient algorithms to perform operations on this class of matrices in order to implement in terms of algorithms the constructive proofs given in Section 2. More precisely, we need to explain how to efficiently perform the following tasks assuming we are given GV representations of $M = D + t(UV^*) = \mathrm{GV}(\mathcal{G}, W, D) \in \mathcal{QSH}_k$ and of $S = D_S + t(uv^*) \in \mathcal{QSH}_1$, where $U, V \in \mathbb{C}^{n\times k}$, $\mathcal{G}$ spans U, $u, v \in \mathbb{C}^n$ and $u = Ux$ for some vector $x \in \mathbb{C}^k$:

1. Compute a GV representation of rank k of $M + S$.
2. Given a unitary upper Hessenberg matrix P, compute a GV representation of rank k of $t(PU(PV)^*)$, and a GV representation of rank 1 of $R = PMP^* - t(PU(PV)^*)$.

We start by analyzing the problem of computing $M + S$. Since $S = D_S + t(Uxv^*)$, in view of Lemma 4.8, we find that $\mathcal{G}^* S$ is lower banded with bandwidth k and so by applying Lemma 4.7 there exists $W_S \in \mathbb{C}^{n\times k}$ and $\widehat{D_S}$ real diagonal such that $S = \mathrm{GV}(\mathcal{G}, W_S, \widehat{D_S})$ is a GV representation of S. Given an algorithm for the computation of W_S it is possible to represent $M + S$ as $\mathrm{GV}(\mathcal{G}, W_S + W, D + D_S)$.

We need to investigate how to actually compute the matrix W_S assuming we are given a GV representation $S = \mathrm{GV}(\mathcal{F}, z, D_S)$ of S. Recall that the i-th column of W_S can be extracted from the components of the vector $\mathcal{G}^*[i+1:]Me_i$, as explained in Lemma 4.7. We can compute the whole matrix W_S at cost $O(nk)$ by following this procedure:

- Compute the last column of W_S by using Lemma 4.7. This is almost cost-free since no rotations are involved, and the only significant element of $W_S e_{n-1}$ is equal to z_{n-1}.

- Compute $W_S e_i$ starting from $W_S e_{i+1}$; this vector can be computed by using some elements in $\mathcal{G}^*[i+1:]Me_i$. In fact, since we are in the 1-quasiseparable case then $\underline{z}_i = z_i e_{i+1}$. So we have

$$\mathcal{G}^*[i+1:]Me_i = \mathcal{G}^*[i+1:]\mathcal{F}[i+1:]\underline{z}_i = z_i \mathcal{G}^*[i+1:]\mathcal{F}[i+1:]e_i.$$

In particular, the only relevant quantity that we need to compute to obtain a representation for the i-th column of M is $\Gamma_i := \mathcal{G}^*[i+1:]\mathcal{F}[i+1:]e_i$. To this end we have

$$\begin{aligned}\Gamma_i &= \mathcal{G}^*[i+1]\mathcal{G}^*[i+2:]\mathcal{F}[i+2:]\mathcal{F}[i+1]e_i\\ &= \mathcal{G}^*[i+1]\mathcal{G}^*[i+2:]\mathcal{F}[i+2:]\left(\alpha e_i + \beta e_{i+1}\right) =\\ &= \mathcal{G}^*[i+1]\left(\beta\Gamma_{i+1} + \alpha e_i\right)\end{aligned}$$

for some α, β such that $\alpha^2 + \beta^2 = 1$.

The above procedure provides a a recursion for the computation of Γ_i for $i = 1, \dots, n-1$. Moreover, it is clear that the computation of Γ_i from Γ_{i+1} only costs $O(k)$ flops and that guarantees that the algorithm can be carried out within the desired cost bound of $O(nk)$. This gives a $O(nk)$ algorithm for computing a k-quasiseparable representation of S.

We investigate now the problem of computing the residual matrix given by Theorem 2.5 and Corollary 2.6 using the GV representation of M and S. We rephrase the proof of Theorem 2.5 in terms of GV representations.

Analyzing the proof of Theorem 2.5 and its corollaries, we can observe that the algorithm can be constructed easily if we are able to compute the residual $R_i = t(F_i UVF_i^*) - F_i t(UV^*)F_i^*$ for a Givens rotation F_i acting on the rows $(i, i+1)$. In the following, we suppose that $i < n-k-1$ so that we do not need to care about "border conditions". However, all the concepts reported are easily extendable to those cases by just adding some care in the process.

Assume we are given D, U, V, W and $\mathcal{G}$ such that $M = D + t(UV^*) = \mathrm{GV}(\mathcal{G}, W, D)$. Observe that we can compute an updated $\widehat{\mathcal{G}}$ such that $\widehat{\mathcal{G}}$ spans F_iU. In fact, we know that $\mathcal{G}^*F_i^*F_iU$ is of the form $\left[\begin{smallmatrix}\times\\0\end{smallmatrix}\right]$ where $\times$ is an appropriate $k \times k$ block. The rotation F_i^* can be passed through the rotations inside $\mathcal{G}^*$ (by properly updating them using the turnover operation) obtaining $\widehat{F}_{i+k}^*\widehat{\mathcal{G}}^* = \mathcal{G}^*F_i^*$. Then, by $\widehat{F}_{i+k}^*\widehat{\mathcal{G}}^*F_iU = \left[\begin{smallmatrix}\times\\0\end{smallmatrix}\right]$, we can conclude that also $\widehat{\mathcal{G}}^*F_iU = \widehat{F}_{i+k}\left[\begin{smallmatrix}\times\\0\end{smallmatrix}\right] = \left[\begin{smallmatrix}\times\\0\end{smallmatrix}\right]$ since $\widehat{F}_{i+k}$ is operating on the null rows.

Moreover, we can check that $\mathrm{GV}(\widehat{\mathcal{G}}, W, D)$ correctly represents the lower part of $t(F_i(D + UV^*)F_i^*)$ on every column but the one with indices $i, i+1$. In fact, the diagonal part of M is left unchanged on the indices different from $i, i+1$. For the rest of the matrix we can distinguish two cases and we do not need to care about D:

- If $j > i+1$, both the left multiplication by F_i and the right multiplication by F_i^* leave unchanged the relevant part of U and V needed for the computation of the portion of the j-th column contained in the lower part of the matrix. Moreover, since in this case $\widehat{\mathcal{G}}[j+1:] = \mathcal{G}[j+1:]$ we conclude that the proposed representation for these columns is valid.

- Also when $j < i$ the right multiplication by F_i^* does not change the j-th column at all. However, the left multiplication by F_i does change the j-th column and we can verify that $\mathrm{tril}(t(F_iUV^*F_i^*), -1)e_j = \mathrm{tril}(F_iUV, -1)e_j$. Recall that, by definition of GV representation, we have

  $$\mathrm{tril}(t(UV^*), -1)e_j = \mathrm{tril}(UV^*, -1)e_j = \mathrm{tril}(M, -1)e_j = \mathcal{G}[j+i:]\underline{w_i}.$$

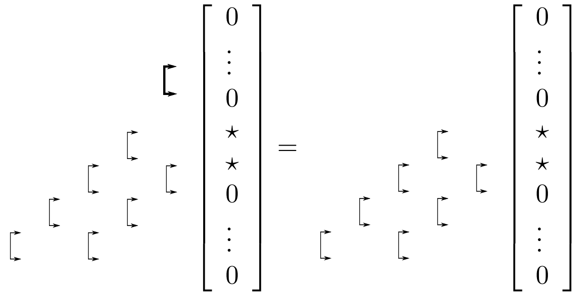
  Since $j < i$ we have $F_i\,\mathrm{tril}(UV^*, -1)e_j = \mathrm{tril}(F_iUV^*, -1)e_j$ so that we can write

  $$\mathrm{tril}(F_iUV^*, -1)e_j = F_i\mathcal{G}[j+1:]\underline{w}_i = \widehat{\mathcal{G}}[j+1:]\widehat{F}_{i+k}\underline{w}_j = \widehat{\mathcal{G}}[j+1:]\underline{w}_j$$

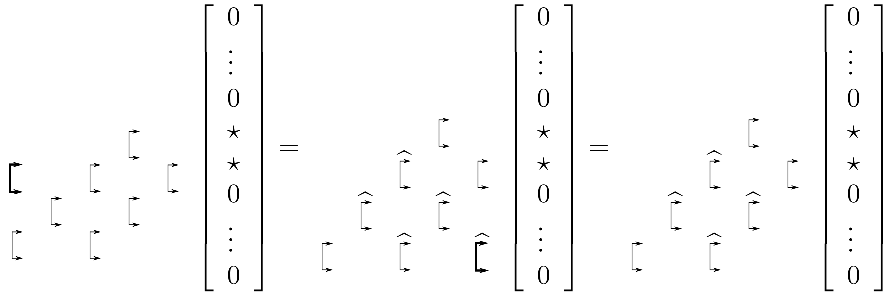
  where the last two equalities follow from the definition of $\widehat{\mathcal{G}}$ and from the fact that the components of $\underline{w}_j$ with index bigger than $j+k$ are zero. Thus we have that $\mathrm{GV}(\widehat{\mathcal{G}}, W, D)$

provides a good representation of the lower part of the j-th column of $t(F_iUV^*F_i^*)$, as requested.

A pictorial representation of these two facts can help to get a better understanding of what is going on (here we are fixing $k = 2$). The rotation F_i on the left is highlighted using the bold font. Equation (14) represents the first case, where the rotation F_i does not intersect the indices of the rotations in $\mathcal{G}[j+1:]$, and Equation (15) the latter case, where an update of the rotations is necessary.

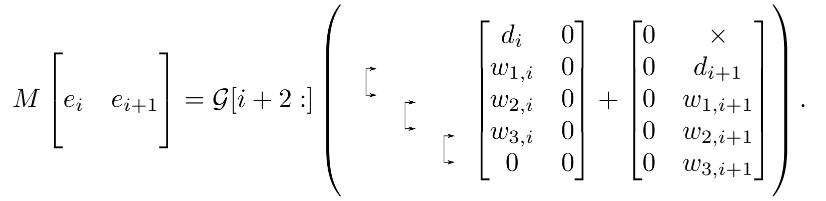


(14)

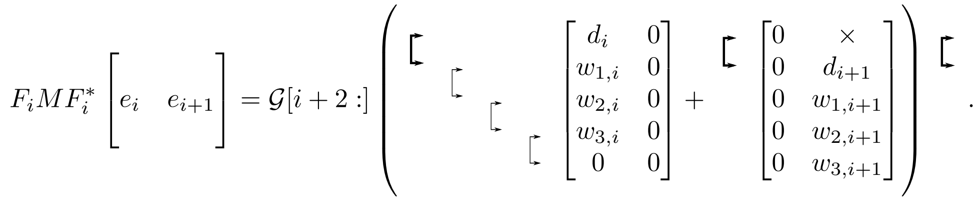


(15)

This means that we need to track only what happens on columns $(i, i+1)$. We show how to update D and W in the i an $i+1$ components in order to account for what happens in these indices. Note that these columns of M can be described in the following way (we report the case $k = 3$ for simplicity):



Left and right multiplying by F_i and F_i^* (reported with the bold font), respectively, leads to the following structure:



We can explicitly compute the value inside the brackets and then observe that, since $\widehat{\mathcal{G}}[i+2:] = \mathcal{G}[i+2:]$, we have a representation of the columns of $F_iMF_i^*$. Now we want to find a Hermitian matrix R of the form $R = \alpha e_{i+1}e_i^t + \overline{\alpha}e_ie_{i+1}^t$, $\widehat{w}_{j,i}$, $\widehat{w}_{j,i+1}$ for $j = 1, \ldots, k$ and $\widehat{d}_i$, $\widehat{d}_{i+1}$ such that, writing with $\widehat{[}$ the rotations taken from $\widehat{\mathcal{G}}$, we have

$$\underbrace{\left([\; [\; [\; [\begin{bmatrix} d_i & 0 \\ w_{1,i} & 0 \\ w_{2,i} & 0 \\ w_{3,i} & 0 \\ 0 & 0 \end{bmatrix} + [\begin{bmatrix} 0 & \times \\ 0 & d_{i+1} \\ 0 & w_{1,i+1} \\ 0 & w_{2,i+1} \\ 0 & w_{3,i+1} \end{bmatrix} \right)}_{C} [\; +R = \widehat{[}\; \widehat{[}\; \widehat{[} \begin{bmatrix} \widehat{d}_i & 0 \\ \widehat{w}_{1,i} & 0 \\ \widehat{w}_{2,i} & 0 \\ \widehat{w}_{3,i} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \times \\ 0 & \widehat{d}_{i+1} \\ 0 & \widehat{w}_{1,i+1} \\ 0 & \widehat{w}_{2,i+1} \\ 0 & \widehat{w}_{3,i+1} \end{bmatrix}. \quad (16)$$

Let C be the left matrix in (16). Then the elements $\widehat{w}_{j,i+1}$ must coincide with the vector $C[3:,2]$, the diagonal elements $\widehat{d}_i$ and $\widehat{d}_{i+1}$ are determined by the diagonal of the top 2×2 block of C. It remains to determine the elements $\widehat{w}_{j,i}$ and the value α. To find them we can multiply on the left by the inverses of the rotations in $\widehat{\mathcal{G}}[i]$. We get the equation

$$\widehat{[}\; \widehat{[}\; \widehat{[} \left(Ce_1 + \begin{bmatrix} 0 \\ \alpha \\ 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} \widehat{d}_i \\ \widehat{w}_{1,i} \\ \widehat{w}_{2,i} \\ \widehat{w}_{3,i} \\ 0 \end{bmatrix}.$$

We can choose α such that we get a 0 in the last component (which can always be done if the rotations are not trivial) and then set the values $\widehat{w}_{j,i}$ by back substitution.

5. Reduction algorithm

In this section we explain how the reduction algorithm can be constructed by using the tools presented in the previous sections.

Recall that the matrix $A = D + UV^*$ can be represented in the more general form $A = t(UW^*) + S + UV^*$, where S is a Hermitian 1-quasiseparable matrix and $U, V, W \in \mathbb{C}^{n \times k}$, just by setting $W = 0$ and $S = D$. Recall also that, by Theorem 2.5, this form is maintained by the trailing principal submatrices $\widehat{A}_j \in \mathbb{C}^{(n-j+1)\times(n-j+1)}$ of the matrices A_j, generated at each step j of the algorithm, that is, $\widehat{A}_j = t(U_jW_j^*) + S_j + U_jV_j^*$. The matrices U_j and V_j are easily obtained by multiplying U_{j-1} and V_{j-1} by a sequence of Givens rotations and by removing the first row. The matrices S_j and W_j will be used to store the "residues".

A high level overview of the algorithm is reported in the pseudo-code of Algorithm 2.

The functions in the code of Algorithm 2 perform the following operations:

`cleanColumn`(v) is a function that takes as input a column vector and returns a sequence of Givens rotations $\mathcal{G}$ such that $\mathcal{G}v = v_1e_1 + \alpha e_2$ for some α.

$(R_M, M) \leftarrow$`conjugateAndTruncate`$(M, \mathcal{G})$ takes as input a quasiseparable Hermitian matrix $M \in \mathcal{QSH}_k$ and a sequence of Givens rotations $\mathcal{G}$. Then it computes a quasiseparable representation for $\mathcal{G}M\mathcal{G}^* - R_M$ where R_M is a matrix in $\mathcal{QSH}_1$. It returns an updated representation of M and the residual matrix R_M.

Algorithm 2 High level reduction process

```
$A_1 \leftarrow D + UV^*$
$M \leftarrow 0$
$S \leftarrow 0$
$s \leftarrow \texttt{zeros}(1, n-1)$
$d \leftarrow \texttt{zeros}(1, n)$
for $i = 1, \dots, n-2-k$ do
    $\mathcal{G} \leftarrow \texttt{cleanColumn}(A_i[:, 1])$
    $d[i] \leftarrow (\mathcal{G}A_i[:, 1])[1]$
    $s[i] \leftarrow (\mathcal{G}A_i[:, 1])[2]$
    $U \leftarrow (\mathcal{G} \cdot U)[2:n, :]$
    $V \leftarrow (\mathcal{G} \cdot V)[2:n, :]$
    $(R_M, M) \leftarrow \texttt{conjugateAndTruncate}(M, \mathcal{G})$
    $(R_S, S) \leftarrow \texttt{conjugateAndTruncate}(S, \mathcal{G})$
    $M \leftarrow M + S$
    $S \leftarrow R_M + R_S$
end for
$(d[n-1-k:n], s[n-1-k:n-1], U, V) \leftarrow \texttt{reduceTrailingBlock}(A_{n-1-k})$
```

$S \leftarrow R_M + R_S$ computes the sum of the matrices $R_M, R_S \in \mathcal{QSH}_1$. This is done by assuming that both have the same sequence of Givens rotations in their representation.

`reduceTrailingBlock`(A) reduces the last $k \times k$ block of the matrix using a standard Hessenberg reduction process. This is done because, in the last steps, the trailing block does not have any particular structure anymore.

Some numerical issues might be encountered in the above version of the algorithm. For instance, some cancellation may happen in the sum $R_M + R_S$, which eventually may affect the Givens rotations of the representation of M.

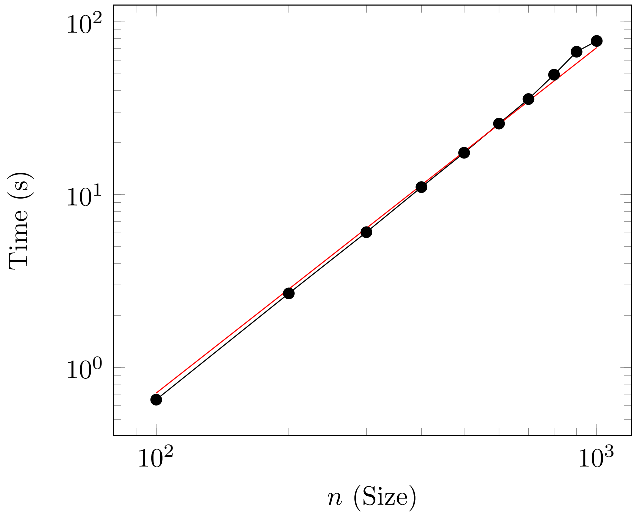
A technique based on re-orthogonalization can be used to restore better approximations. Recall that the rotations inside the GV representation of M are such that $\mathcal{G}^*U = \left[\begin{smallmatrix} Z \\ 0 \end{smallmatrix}\right]$. Such rotations are not unique but (at least with some hypothesis on irreducibility) are essentially unique, that is, they can be determined up to a multiplicative constant of modulus 1. Based on this information we can compute rotations in order to obtain $\mathcal{G}^*U$ in the desired form and correct the moduli of the sine and cosine inside $\mathcal{G}$ without altering the signs.

This has shown to be quite effective in practice, leading to better numerical results. The cost of a reorthogonalization is the cost of a QR factorization of U, thus asymptotically $O(nk^2)$. By performing it every k steps we have a total cost of the modified reduction algorithm that is still $O(n^2k)$, since we need $O(n)$ steps to complete the reduction and $O(nk) \cdot \frac{1}{k}O(n) = O(n^2k)$ ops.

6. Numerical experiments

The algorithm presented in Section 5 has been implemented in the Julia language. It has been run on a Laptop with an Intel(R) Core(TM) i3-2367M CPU running at 1.40GHz and 4 GB of RAM.

In order to analyze the complexity and the accuracy of the results we have performed the following tests:



Size	Time (s)
100	0.65
200	2.68
300	6.06
400	11.05
500	17.5
600	25.8
700	35.7
800	49.4
900	67.17
1000	77.56

Figure 1: CPU time, in seconds, for the Hessenberg reduction of a diagonal plus rank 10 matrix of size n. Here the line is the plot of γn^2 for an appropriate γ. It is evident the quadratic behavior of the time.

- We have run the algorithm on matrices of different sizes but with constant quasiseparability rank $k = 10$. The purpose is to verify that the CPU time is quadratic in the dimension of the problem.

- We have run the algorithm at a fixed dimension $n = 200$ with values of k between 5 and 160. Here the goal is to verify that the CPU time grows linearly with the rank.

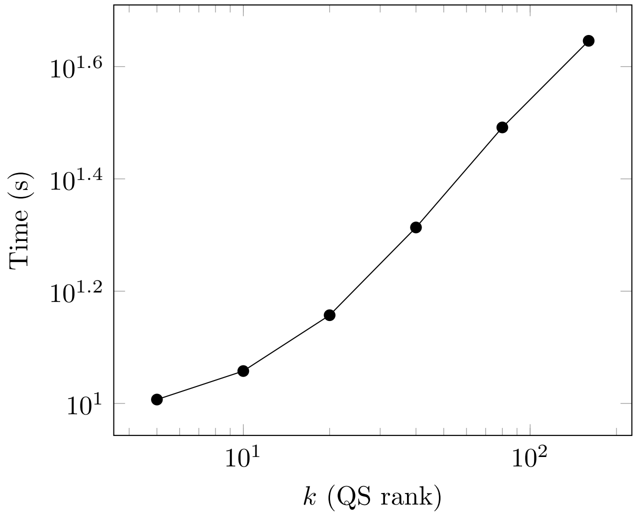
- We have run the algorithm on some test problems in order to measure the errors on the eigenvalues computed starting from the final Hessenberg form. The purpose of this set of tests is to check the numerical stability of the algorithm.

Every experiment has been run 10 times and the mean value of the timings has been taken. In Figure 1 we have reported, in log scale, the timings for some experiments with $n = 100 \cdot i$ for $i = 1, \dots, 10$. In Figure 2 we have reported the CPU time in the case of matrices of fixed size $n = 400$ with various quasiseparable ranks ranging from 5 to 160.

Looking at the results in Figure 2 we see that the complexity in the rank is almost sublinear at the start. This is due to the inefficiency of operations on small matrices and the overhead of these operations in our Julia implementation. The linear trend starts to appear for larger ranks.

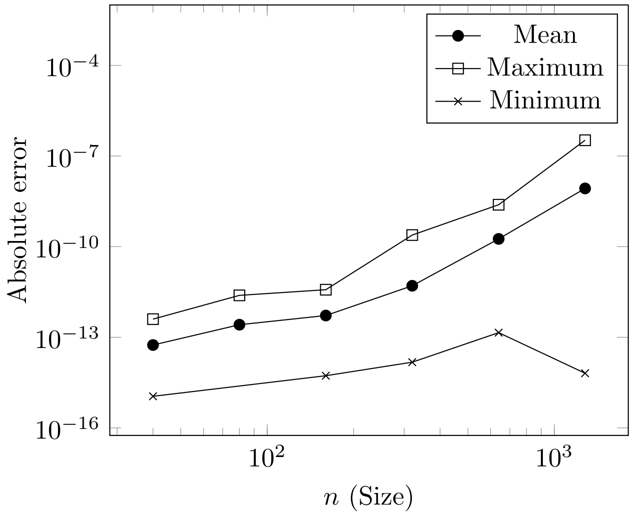
As a last experiment in Figure 3 and Figure 4 we have reported the absolute and relative errors, respectively, on eigenvalue computations for various sizes and fixed quasiseparable rank. The errors were obtained as differences between the eigenvalues computed from the starting full matrix using the QR algorithm and the QR algorithm applied to the Hessenberg matrix provided by our algorithm. In these examples the re-orthogonalization technique described in Section 5 has been used, in order to mitigate the errors.

The matrices in these examples have been obtained by using the `randn` function that constructs matrices whose elements are drawn from a $N(0, 1)$ Gaussian distribution. This function has been used to construct D, U and V diagonal and $n \times k$, respectively, such that $A = D + UV^*$.



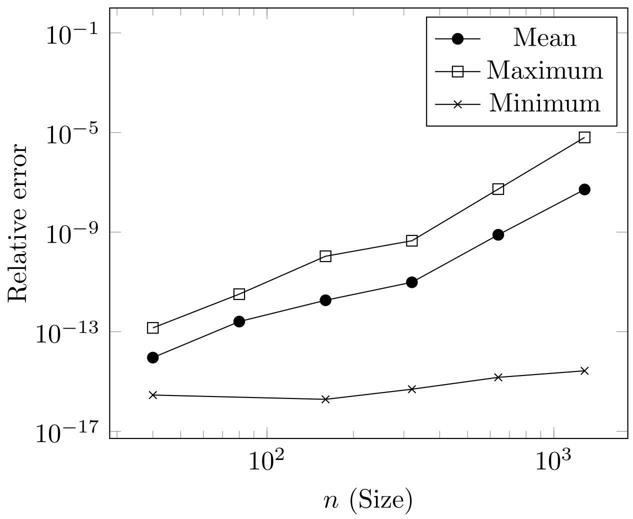
QS Rank	Time (s)
5	10.16
10	11.42
20	14.36
40	20.58
80	31.02
160	44.25

Figure 2: CPU time, in seconds, for the Hessenberg reduction of a 400 × 400 diagonal plus rank k matrix.



Size	Mean	Minimum	Maximum
40.0	5.52e-14	1.11e-15	3.97e-13
80.0	2.59e-13	0.0	2.43e-12
160.0	5.23e-13	5.32e-15	3.74e-12
320.0	5.06e-12	1.49e-14	2.41e-10
640.0	1.80e-10	1.42e-13	2.43e-9
1280.0	8.43e-9	6.43e-15	3.32-7

Figure 3: Absolute errors on eigenvalues computation for random matrices of quasiseparable rank 30 and variable sizes.



Size	Mean	Minimum	Maximum
40.0	9.13e-15	2.86e-16	1.42e-13
80.0	2.55e-13	0.0	3.22e-12
160.0	1.83e-12	1.92e-16	1.07e-10
320.0	9.66e-12	4.91e-16	4.49e-10
640.0	7.79e-10	1.47e-15	5.34e-8
1280.0	5.15e-8	2.69e-15	6.35e-6

Figure 4: Relative errors on eigenvalues computation for random matrices of quasiseparable rank 30 and variable sizes.

7. An application

Let $P(x) = \sum_{i=0}^{d} P_i x^i$ be a matrix polynomial where P_i are $k \times k$ matrices. In [2] a companion linearization A for $P(x)$ has been introduced where A is an $n \times n$ matrix, $n = dk$, of the form (1) with $n = dk$. The computation of the eigenvalues of $P(x)$, that is, the solutions of the equation $\det P(x) = 0$, is therefore reduced to solving the linear eigenvalue problem for the matrix A. The availability of the Hessenberg form H of A with the quasiseparability structure, enables one to apply the QR iteration to H at a low cost by exploiting the both the Hessenberg and the quasiseparable structures.

A different approach to solve the equation $\det P(x) = 0$, followed in [1], consists in applying the Ehrlich-Aberth iteration to the polynomial $\det P(x)$, or alternatively, to represent the polynomial $P(x)$ in secular form [3]. In this case, one has to compute $\det P(x)$ at different values $x_1, \ldots, x_n$. Indeed, the evaluation of $\det P(x)$ at a single value $x = \xi$ can be performed by applying the Horner rule in order to compute the matrix $P(\xi)$ and then by applying Gaussian elimination for computing the determinant $\det P(\xi)$. The overall cost is $O(dk^2 + k^3)$ so that the computation at $n = dk$ different points has the cost $O(d^2k^3 + dk^4)$.

The availability of the structured Hessenberg form H allows us to reduce this cost. We will show that computing $\det(xI - A) = \det(xI - H)$ can be performed in $O(nk) = O(dk^2)$ operations. Asymptotically, the value of this cost is the minimum that we can obtain. In fact, the matrix A is defined by $2nk + n$ entries, so that any algorithm which takes in input these $2nk + n$ values must perform at least $nk + n/2$ operations. In fact each pair of input data must be involved in at least an operation. In the case where we have to compute $\det(xI - A)$ at dk different values we obtain the cost $O(d^2k^3)$ which improves the cost required by the Horner rule applied to the matrix polynomial.

The algorithm for computing $\det(xI - H)$ at a low cost relies on the Hyman method [14]. Assume that H is in k-quasiseparable Hessenberg form. Consider the system $(xI - H)v = \alpha e_1$, set $v_n = 1$ so that the equations from the second to the last one form a triangular k-quasiseparable system. The first equation provides the value of α after that $v_1, \ldots, v_{n-1}$ have been computed. Since a triangular quasiseparable system can be solved with $O(nk)$ ops, we are able to compute α at the same cost. On the other hand, by the Cramer rule we find that $1 = v_n = \alpha(\prod_{i=2,n} h_{i,i-1})/\det(xI - H)$ which provides the sought value of $\det(xI - H)$. A similar approach can be used for computing the Newton correction $p(x)/p'(x)$ for $p(x) = \det(xI - H)$.

Acknowledgments

We wish to thank the referees for providing useful comments and remarks that helped to improve the paper.

References

[1] Bini, D. A., Noferini, V., 2013. Solving polynomial eigenvalue problems by means of the Ehrlich-Aberth method. Linear Algebra Appl. 439, 1130–1149.

[2] Bini, D. A., Robol, L., 2014. On a class of matrix pencils equivalent to a given matrix polynomial. arXiv preprint arXiv:1406.1025.

[3] Bini, D. A., Robol, L., 2014. Solving secular and polynomial equations: a multiprecision algorithm. J. Comput. Appl. Math. 272, 276–292.
URL `http://dx.doi.org/10.1016/j.cam.2013.04.037`

[4] Eidelman, Y., Gohberg, I., Gemignani, L., 2007. On the fast reduction of a quasiseparable matrix to Hessenberg and tridiagonal forms. Linear algebra and its applications 420 (1), 86–101.

[5] Eidelman, Y., Gohberg, I., Haimovici, I., 2014. Separable type representations of matrices and fast algorithms. Vol. 1. Vol. 234 of Operator Theory: Advances and Applications. Birkhäuser/Springer, Basel, basics. Completion problems. Multiplication and inversion algorithms.

[6] Eidelman, Y., Gohberg, I., Haimovici, I., 2014. Separable type representations of matrices and fast algorithms. Vol. 2. Vol. 235 of Operator Theory: Advances and Applications. Birkhäuser/Springer Basel AG, Basel, eigenvalue method.

[7] Eidelman, Y., Gohberg, I., Olshevsky, V., 2005. The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order. Linear Algebra Appl. 404, 305–324.
URL `http://dx.doi.org/10.1016/j.laa.2005.02.037`

[8] Frederix, K., Delvaux, S., Van Barel, M., 2013. An algorithm for computing the eigenvalues of block companion matrices. Numerical Algorithms 62 (2), 261–287.

[9] Smith, B. T., 1970. Error bounds for zeros of a polynomial based upon Gerschgorin's theorems. J. Assoc. Comput. Mach. 17, 661–674.

[10] Van Camp, E., 2005. Diagonal-plus-semiseparable matrices and their use in numerical linear algebra. Ph.D. thesis, PhD thesis, Katholieke Universiteit Leuven.

[11] Vandebril, R., Van Barel, M., Mastronardi, N., 2005. An implicit QR algorithm for symmetric semiseparable matrices. Numer. Linear Algebra Appl. 12 (7), 625–658.
URL `http://dx.doi.org/10.1002/nla.425`

[12] Vandebril, R., Van Barel, M., Mastronardi, N., 2008. Matrix computations and semiseparable matrices. Eigenvalue and singular value methods. Vol. 2. Johns Hopkins University Press, Baltimore, MD.

[13] Vandebril, R., Van Barel, M., Mastronardi, N., 2008. Matrix computations and semiseparable matrices. Linear systems. Vol. 1. Johns Hopkins University Press, Baltimore, MD.

[14] Wilkinson, J. H., 1965. The Algebaic Eigenvalue Problem. Oxford University Press.