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

Keywords: Quasiseparable matrices, Polynomial eigenvalue problems, Hessenberg form.

1. Introduction. Reducing an \( n \times n \) matrix to upper Hessenberg form by a unitary similarity 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 \((k_1,k_2)\)-quasiseparable if its submatrices contained in the strictly upper triangular part have rank at most \( k_2 \) and those contained in the strictly lower triangular part have rank at most \( k_1 \). For simplicity, if \( k_1 = k_2 = k \) 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 \( \alpha \) is strictly greater than 1, in particular \( \alpha = 3 \) for the algorithm of [4]. 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 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 \( n^\delta \) these algorithms have cost \( O(n^3) \) as for a general unstructured matrix.

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

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

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 [2], [3]. They have been also used 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.

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 Theorems 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 (1.1) 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 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 quasiseparable matrix can be easily expressed by using the MATLAB notation. In fact, $A$ is lower-quasiseparable of rank $k$ if and only if

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

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

2
Given a vector \( v = (v_1, v_2) \in \mathbb{C}^2 \), denote by \( G = G(v_1, v_2) \) a 2 × 2 Givens rotation
\[
G = \begin{bmatrix}
    c & s \\
    -\bar{s} & c
\end{bmatrix}, \quad c \in \mathbb{R}, \quad |s|^2 + c^2 = 1,
\]
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 \( G = \begin{bmatrix} \frac{s_i}{c_i} & s_i \\ -\bar{s}_i & c_i \end{bmatrix} \) to the components \( i \) and \( i + 1 \) of a vector, where \( I_n \) denotes the identity matrix of size \( n \) and \( A \oplus B \) denotes the block diagonal matrix \( \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \). With abuse of terminology, we will call \( G_i \) Givens rotations as well.

The following well-known result \cite{13} will be used in our analysis:

**Lemma 2.1.** Let \( Q \) be a unitary Hessenberg matrix. Then
- \( Q \) is a \((1,1)\)-quasiseparable matrix.
- \( Q \) can be factorized as a product of \( n - 1 \) Givens rotations \( Q = G_1 \cdots G_{n-1} \).

In the following we use the operators \( \text{tril}(\cdot, \cdot) \) and \( \text{triu}(\cdot, \cdot) \), coherently with the corresponding MATLAB functions, such that \( L = \text{tril}(A, k) \), \( U = \text{triu}(A, k) \) where \( A = (a_{i,j}) \), \( L = (l_{i,j}) \), \( U = (u_{i,j}) \)
\[
l_{i,j} = \begin{cases} a_{i,j} & \text{if } i \geq j - k \\ 0 & \text{otherwise} \end{cases} \quad u_{i,j} = \begin{cases} a_{i,j} & \text{if } i \leq j - k \\ 0 & \text{otherwise} \end{cases}.
\]

**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
\[
t(A) = \text{tril}(A, -1) + \text{triu}(A^*, 1), \tag{2.1}
\]
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 \( \mathcal{QSH}_k^n \) 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
\[
t(A + B) = t(A) + t(B), \quad \text{for any } A, B \in \mathbb{C}^{n \times n}, \tag{2.2}
\]
\[
t(\alpha A) = \alpha t(A), \quad \text{for any } \alpha \in \mathbb{R},
\]
\[
t(DAD^*) = Dt(A)D^*, \quad \text{for any } D \text{ diagonal matrix},
\]
moreover,
\[
t \left( \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \right) = \begin{bmatrix} t(A_{1,1}) & A_{2,1}^* \\ A_{1,2} & t(A_{2,2}) \end{bmatrix}, \tag{2.3}
\]
where \( A_{1,1} \) and \( A_{2,2} \) are square matrices. We also have
\[
t(A) = A - \text{diag}(a_{1,1}, \ldots, a_{n,n}), \quad \text{for any } A \text{ such that } A = A^*. \tag{2.4}
\]

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 2.2.** 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
\[
t(SAS^*) - St(A)S^* = W \oplus 0_{n-k},
\]
\[
3
\]
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 (2.3) we have

$$t((SAS^*)^*) = \begin{bmatrix} t(ZA_{1,1}Z^*) & ZA_{2,1} \\ A_{2,1}Z^* & t(A_{2,2}) \end{bmatrix}. \quad (2.5)$$

On the other hand,

$$St(A)S^* = S \begin{bmatrix} t(A_{1,1}) & A_{2,1}^* \\ A_{2,1} & t(A_{2,2}) \end{bmatrix} S^* = \begin{bmatrix} Zt(A_{1,1})Z^* & ZA_{2,1}^* \\ A_{2,1}Z^* & t(A_{2,2}) \end{bmatrix}. \quad (2.6)$$

So that, from (2.5) and (2.6) 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 $t(DAD^*) - St(D)D^* = 0$ in view of (2.2). \ \square

Lemma 2.3. Let $S = (Z \oplus I_{n-2})(1 \oplus \hat{S})$ where $Z \in \mathbb{C}^{2 \times 2}$, $\hat{S} \in \mathbb{C}^{(n-1) \times (n-1)}$. Then for any matrix $A$ partitioned as $A = \begin{bmatrix} A_{1,1} & v^* \\ v & \lambda \end{bmatrix} \in \mathbb{C}^{n \times n}$, where $\lambda \in \mathbb{C}^{(n-1) \times (n-1)}$ it holds that

$$t((SAS^*)^*) - St(A)S^* = W \oplus 0_{n-2} + (Z \oplus I_{n-2})(0 \oplus (t(\hat{S}\hat{S}^*) - \hat{S}t(\hat{A})\hat{S}^*)'(Z^* \oplus I_{n-2}).$$

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

Proof. Set $B = (1 \oplus \hat{S})A(1 \oplus \hat{S}^*)$ then by Lemma 2.2

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

On the other hand

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

Thus

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

where $E = t(B) - (1 \oplus \hat{S})t(A)(1 \oplus \hat{S})$. Now, since $B = (1 \oplus \hat{S})A(1 \oplus \hat{S})^*$, in view of (2.3) we have

$$B = \begin{bmatrix} A_{1,1} & v^* \hat{S}^* \\ \hat{S}v & \hat{S}\hat{S}^* \end{bmatrix}, \quad t(B) = \begin{bmatrix} 0 & v^* \hat{S}^* \\ \hat{S}v & t(\hat{S}\hat{S}^*) \end{bmatrix}.$$ 

A similar analysis shows that

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

Thus we get

$$t((SAS^*)^*) - St(A)S^* = W \oplus 0_{n-2} + (Z \oplus I_{n-2})(0 \oplus (t(\hat{S}\hat{S}^*) - \hat{S}t(\hat{A})\hat{S}^*)'(Z^* \oplus I_{n-2}).$$
A consequence of the above results is expressed by the following

**Theorem 2.4.** Let $A \in \mathbb{C}^{n \times n}$, and set $Q = G_h \cdots G_k$ for $1 \leq h < k \leq n - 1$, where the parameters $s_i, c_i$ defining $G_i$ are such that $s_i \neq 0$, $i = h, \ldots, k$. Then $R_n := t(QAQ^*) - QT(A)Q^* = \text{diag}(d) + t(ab^*) \in QSH_1^+$ for vectors $a, b, d \in \mathbb{C}^n$, where $b$ is independent of $A$. More precisely, $b_k = s_k \cdots s_h, b_i = c_{i-h} s_i \cdots s_h$, for $i = h + 1, \ldots, k, b_i = a_i = d_i = 0$ 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 2.2 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 $i = 1, \ldots, h - 1$ and for $i = k + 2, \ldots, n$ 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 $i = 1, \ldots, n - 1$. We prove that $R_n \in QSH_1^+$ by induction on $n$. For $n = 2$ it holds that $R_2 = [0 \ 1 \ 1]$. 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 $Q = (Z \oplus I_{n-2})(1 \oplus \tilde{S})$ for $Z = [\begin{smallmatrix} \alpha & \ast \\ \ast & \ast \end{smallmatrix}]$, and $\tilde{S} \in \mathbb{C}^{(n-1) \times (n-1)}$, where for notational simplicity we set $s = s_1, c = c_1$. Applying Lemma 2.3 yields

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

for

$$W = t(Z[\begin{smallmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{smallmatrix}] Z^*) - Zt([\begin{smallmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{smallmatrix}])Z^*$$

$$= \begin{bmatrix} \alpha & c_{a_{2,1}} + s_{a_{1,2}} \sigma_1 \\ -s_{a_{2,1}} - c_{a_{2,1}} \sigma_1 & -s_{a_{1,2}} - c_{a_{1,2}} \sigma_1 \end{bmatrix},$$

where $R_{n-1} = t(\tilde{S}_1 \tilde{A} \tilde{S}_1^*) - \tilde{S}_1 (\tilde{A}) \tilde{S}_1^*$ and $\tilde{A}$ is the trailing principal submatrix of $A$ of size $n - 1$. A direct inspection shows that

$$R_n = W \oplus 0_{n-2} + \begin{bmatrix} |s|^2 e_1^T R_{n-1} e_1 \\ \bar{e}_1^T D R_{n-1} e_1 \end{bmatrix} D = c \oplus I_{n-2}. \quad (2.7)$$

From the inductive assumption we may write that $R_{n-1} = \text{diag}(\bar{d}) + t(\bar{a} \bar{b}^*)$ for $\bar{a}, \bar{b}, \bar{d} \in \mathbb{C}^{n-1}$, where $\bar{b}_1 = s_2 \cdots s_{n-1} \neq 0, \bar{b}_1 = c_{i-s_{i+1}} \cdots s_{n-1}$. So that (2.7) turns into

$$R_n = \begin{bmatrix} \alpha_1 & \alpha_{1,2} \\ \alpha_{2,1} & \alpha_{2,2} \end{bmatrix} + \begin{bmatrix} |s|^2 \hat{d}_1 \\ \alpha_2 b_{1} \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_{1,2} \\ \alpha_{2,1} & \alpha_{2,2} \end{bmatrix} + \begin{bmatrix} \alpha_2 b_{1} \end{bmatrix}$$

where the upper triangular part, denoted with $\ast$, is determined by symmetry. Thus, it follows that $R_n = \text{diag}(d) + t(a,b)$ where $d_1 = |s_1|^2 \hat{d}_1 + w_{1,1}, d_2 = c_2 \hat{d}_1 + w_{2,2}, d_i = \hat{d}_{i-1}$, for $i = 3, \ldots, n$; moreover

$$a_2 = \frac{1}{b_1} (c \bar{a}_d + 2 c_2 \bar{d}_1 + w_{2,1}),$$
$$a_i = \bar{a}_{i-1}, \text{ for } i = 3, \ldots, n,$$
$$b_1 = s_1 \hat{b}_1, \quad b_2 = c_1 \hat{b}_1,$$
$$b_i = \hat{b}_{i-1}, \text{ for } i = 3, \ldots, n - 1.$$
where the condition \( s_i \neq 0 \) implies that \( b_1 \neq 0 \). This completes the proof. \( \square \)

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

**Corollary 1.** In the hypotheses of Theorem 2.4, we have

\[
R = t(QAQ^*) - Q(D + t(A))Q^* \in QSH_n^b
\]

for any diagonal matrix \( D \) with real entries. Moreover, \( R = \text{diag}(d) + t(ab^*) \) 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 2.4, \( t(QAQ^*) - Q((A))Q^* = \text{diag}(d) + t(ab^*) \), where \( b \) is independent of \( A \). On the other hand, equation (2.4) implies that \( QDQ^* = \text{diag}(d') + t(QDQ^*) \), moreover, since \( t(D) = 0 \), Theorem 2.4 implies that \( t(QDQ^*) = t(ab^*) \) for some vector \( a \). Therefore \( R = \text{diag}(d) + t(ab^*) - \text{diag}(d') = \text{diag}(d + d') + t(\langle a - a'\rangle b^*) \), that is, \( R \in QSH_n^b \). \( \square \)

A further analysis enables us to provide an explicit representation of the \( i \)th row of the matrix \( R \). More precisely we have the following result:

**Theorem 2.5.** Under the assumptions of Theorem 2.4, the \( i \)th row of the matrix \( R = t(QAQ^*) - Q(D + t(A))Q^* \) has the representation

\[
e_i^TR = [0, \ldots, 0, v_i, d_i, w_i^*]G_{i-2}^*G_{i-3}^* \cdots G_1^* \quad (2.8)
\]

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 \( Q = Q_1Q_2 \) where \( Q_1 = G_1 \cdots G_{i-2}, Q_2 = G_{i-1} \cdots G_{n-1} \), so that (2.8) can be rewritten as \( e_i^T RQ_1 = [0, \ldots, 0, v_i, d_i, w_i^*] \). This way, it is enough to show that the \( j \)th row of \( RQ_1 \) has the first \( j - 2 \) entries equal to zero. In view of Lemma 2.2 we have

\[
R' := t(Q_2AQ_2^*) - Q_2(D + t(A))Q_2^* = 0_{n-2} \oplus \tilde{R} \in QSH_n^b.
\]

Whence

\[
Q_2(D + t(A))Q_2^* = t(Q_2AQ_2^*) - 0_{n-2} \oplus \tilde{R}. \quad (2.9)
\]

Moreover, by definition of \( R \) we have

\[
RQ_1 = t(Q_1Q_2AQ_2^*Q_1^*)Q_1 - Q_1Q_2(D + t(A))Q_2^*.
\]

Setting \( B = Q_2AQ_2^* \) and combining the above equation with (2.9) yields

\[
RQ_1 = t(Q_1BQ_1^*)Q_1 - Q_1t(B) + Q_1(0_{n-2} \oplus \tilde{R}).
\]

Now, since \( Q_1(0_{n-2} \oplus \tilde{R}) \) has the first \( i - 2 \) columns equal to zero, it is sufficient to prove that the \( i \)th row of \( t(Q_1BQ_1^*)Q_1 - Q_1t(B) \) has the first \( i - 2 \) component zero. To this regard, observe that \( Q_1 = \hat{Q}_1 \oplus I_{n-i+1}, \) where \( \hat{Q}_1 \in \mathbb{C}^{(i-1) \times (i-1)} \), so that partitioning \( B \) as \( [B_{1,1} \ B_{1,2} \ B_{2,1} \ B_{2,2}] \), where \( B_{1,1} \in \mathbb{C}^{(i-1) \times (i-1)} \), by applying again Lemma 2.2 we find that \( t(Q_1BQ_1^*) - Q_1t(B)Q_1^* = W_{i-1} \oplus 0_{n-i+1} \) for some \( W_{i-1} \in \mathbb{C}^{(i-1) \times (i-1)} \). This implies that \( t(Q_1BQ_1^*)Q_1 - Q_1t(B) \) has the last \((n-i+1) \) rows equal to zero. This completes the proof. \( \square \)

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 lower triangular part of $H$ have rank at most 1 so that the submatrices in the lower triangular part of $S = QDQ^*$ have rank at most $k$. On the other hand, since $S$ is Hermitian, then also the submatrices contained in its upper triangular part have rank at most $k + 1$, thus the submatrices in the 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 later in Section 5.3 that the submatrices contained in the 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. In order to arrive at this goal we have to recall the customary procedure for reducing a matrix $A$ into Hessenberg form which is based on Givens rotations. The algorithm can be easily described by the pseudo-code reported in Algorithm 1 where the function $\text{givens}(v_1, v_2)$ provides the matrix $G(v_1, v_2)$.

Algorithm 1 Reduction to Hessenberg form by means of Givens rotations

1: for $j = 1, \ldots, n - 2$ do
2:    for $i = n, \ldots, j + 2$ do
3:        $G \leftarrow \text{givens}(A[i - 1, j], A[i, j])$
4:        $A[i - 1 : i, ;] \leftarrow G \cdot A[i - 1 : i, ;]$
5:        $A[i ;, i - 1 : i] \leftarrow A[i ;, i - 1 : i] \cdot G^*$
6:    end for
7: end for

Denote by $G_{i-1,j}$ the unitary $n \times n$ matrix which performs the Givens rotation $G = \text{givens}(A[i - 1, j], A[i, j])$ 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

$$A_0 = A, \quad A_{n-2} = H$$

$$A_j = Q_j A_{j-1} Q_j^*, \quad j = 1, \ldots, n - 2$$

where the matrix $A_j$ has the form

$$A_j = Q_j \cdots Q_1 (D + UV^*) Q_1^* \cdots Q_j^* = \hat{D} + \hat{U} \hat{V}^* = \begin{bmatrix}
    a_{1,1}^{(j)} & \cdots & a_{1,j}^{(j)} & \times & \cdots & \times \\
    a_{2,1}^{(j)} & \ddots & \vdots & \ddots & \ddots & \ddots \\
    \vdots & \ddots & a_{j,j}^{(j)} & \times & \cdots & \times \\
    a_{j+1,j}^{(j)} & \cdots & * & \ddots & \ddots & \ddots \\
    \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
    * & \ddots & \ddots & \ddots & \ddots & \ddots 
\end{bmatrix}, \quad (3.1)$$

and the symbols $\times$ and $\ast$ denote any numbers.

For notational simplicity, we denote by $\hat{A}_j$ the $(n - j) \times (n - j)$ trailing principal submatrix of $A_j$, that is, the submatrix represented by $\ast$ in (3.1). Observe that $\hat{A}_j$ 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 $i + 1$. 

7
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 \( G_1, \ldots, G_{n-1} \) constructed in such a way that \((G_1 \cdots G_{n-1})v \neq (w^{(1)}*0, \ldots, 0)^*\), where \( w^{(i)} \in \mathbb{C}^i \), for \( i = 1, \ldots, n-1 \). If there exists \( h \) such that \( G_h = I \) then one can choose \( G_i = I \) for every \( i \geq h \), that is, \((G_1 \cdots G_{n-1})v = (w_1^{(1)}*0, \ldots, 0)^*\).

**Proof.** Since \( G_1 \cdots G_{n-1} \) 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 \((G_1 \cdots G_{n-1})v[i : n]\), that is, \( |w_i^{(n)}| \). On the other hand, if \( G_h = I \) then \((w^{(h)}*0, \ldots, 0) = (w^{(h+1)}*0, \ldots, 0)\) so that \( w_{h+1}^{(h+1)} = 0 \). This implies that \( |v[h + 1 : n]| = 0 \), whence \( v_i = 0 \) for \( i = h + 1, \ldots, n \). This way, one can choose \( G_i = I \) for \( i = h + 1, \ldots, n - 1 \). \( \square \)

Observe that in view of Lemma 2.1 the unitary matrix \( Q = G_2 \cdots G_{n-1} \) is in upper Hessenberg form. Moreover, the rotations \( G_i \) are such that \( QAe_1 = \alpha e_1 + \beta e_2 \), where \( \alpha, \beta \in \mathbb{C} \). In view of Lemma 3.1 we can assume that if \( G_i = I \) then \( G_i = I \) for every \( i \geq h \), that is, \( Q = G_2 \cdots G_{h-1} \).

The quasipseparable structure of the matrices \( \hat{A} \) is a consequence of our main result which is reported in the following

**Theorem 3.2.** Let \( U, V, W \in \mathbb{C}^{n \times k} \), \( S = \text{diag}(d) + t(ab^*) \in \mathbb{QSH}_1^n \) 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
\[
QAe_1 = \alpha_1 e_1 + \beta e_2,
\]
where \( Q = G_2 \cdots G_{n-1} \).

Then the matrix \( \hat{A} \) obtained by removing the first row and the first column of \( QAQ^* \) can be written again as
\[
\hat{A} = \tilde{U} \tilde{V}^* + t(\tilde{U} \tilde{W}^*) + \tilde{S},
\]
where \( \tilde{U}, \tilde{W} \in \mathbb{C}^{(n-1) \times k} \), and \( \tilde{S} = \text{diag}(d + t(\tilde{a}\tilde{b}^*)) \in \mathbb{QSH}_1^{n-1} \) for some vectors \( \tilde{a}, \tilde{\varphi}, \tilde{b} \in \mathbb{C}^{n-1} \). Moreover, \( \tilde{U} \) and \( \tilde{V} \) are obtained by removing the first row of \( U \) and \( V \), 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 \( h \leq n - 1 \). In view of Lemma 2.2 without loss of generality we may assume that \( h = n - 1 \). We have
\[
QAQ^* = (QU)(QV)^* + F, \quad F = Q(t(UW^*) + S)Q^*.
\]
In view of Corollary 3 we have \( F = t(Q(UW^* + ab^*)Q^*) - R \) for \( R \in \mathbb{QSH}_1^n \). Thus
\[
QAQ^* = (QU)(QV)^* + t(QU(QW)^*) + t(Qa(b)b^*) - R. \tag{3.2}
\]
Recall from Theorem 2.4 that \( Rc_1 = 0 \) and that \( Qa_c \) has been chosen so that \( QAQ^*e_1 = \alpha e_1 + \beta e_2 \). This fact, together with (3.2), implies that the vector \( u = t(Qa(b)b^*)c_1 \) is such that \( u[3 : n] \) is in the span of the columns of \((QU)[3 : n, :]\). In view of (2.3) we may write \( u = t(Qa(b)b^*)[2 : n, 2 : n] = t(\tilde{u}z^*) \) for \( \tilde{u} = u[2 : n] \), and for a suitable \( z \in \mathbb{C}^{n-1} \). Applying (2.2) yields the following representation for the trailing principal submatrix \( \hat{A} \) of \( QAQ^* \) of size \( n - 1 \)
\[
\hat{A} = \tilde{U} \tilde{V}^* + t(\tilde{U} \tilde{W}^* + \tilde{u}z^*) - \tilde{R},
\]
where \( \tilde{U}, \tilde{V} \) and \( \tilde{W} \) are obtained by removing the first row of \( U, V \) and \( W \), respectively, while \( \tilde{R} = R[2 : n, 2 : n] \). Since \( \tilde{u}[2 : n] \) is in the span of \( \tilde{U}[2 : n, :] \), and since the first row of \( \tilde{U} \) as well
as the first entry of \( \hat{\mathbf{u}} \) do not play any role in the value of \( t(\hat{\mathbf{U}}\hat{\mathbf{W}}^* + \hat{\mathbf{u}}\hat{\mathbf{z}}^*) \), we may set \( \hat{\mathbf{u}}_1 \) equal to an appropriate value in such a way that \( \hat{\mathbf{u}} \) is in the span of the columns of \( \hat{\mathbf{U}} \). This way, the matrix \( \hat{\mathbf{U}}\hat{\mathbf{W}}^* + \hat{\mathbf{u}}\hat{\mathbf{z}}^* \) has rank at most \( k \) and can be written as \( \hat{\mathbf{U}}\hat{\mathbf{W}}^* \) for a suitable \( \hat{\mathbf{W}} \in \mathbb{C}^{k \times n} \). Thus we have

\[
\hat{\mathbf{A}} = \hat{\mathbf{U}}\hat{\mathbf{W}}^* + t(\hat{\mathbf{U}}\hat{\mathbf{W}}^*) + \hat{\mathbf{S}}
\]

for \( \hat{\mathbf{S}} = -\hat{\mathbf{R}} \), that concludes the proof. \( \square \)

Observe that the matrix \( \hat{\mathbf{A}} \) defined in (1.1) 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 \( \hat{\mathbf{A}}_j \) of the sequence generated by the Hessenberg reduction, maintains the structure \( \hat{\mathbf{A}}_1 = U_jV_j^* + t(U_jW_j^*) + S_j \) where \( S_j = \text{diag}(d_j) + t(a_jb_j^*). \)

This fact is fundamental to design fast algorithms for the Hessenberg reduction of a matrix of the form (1.1). In order to realize this goal, we need to choose a reliable explicit representation for \( t(U_jW_j^*) \) and \( S_j \). This is the topic of the next section.

Note also that the representation \( \hat{\mathbf{A}}_j = U_jV_j^* + t(U_jW_j^*) + S_j \) implies that \( \hat{\mathbf{A}}_j \) is still a quasiseparable matrix of low quasiseparability rank. In fact, \( t(U_jW_j^*) + U_jV_j^* \) 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 \( \hat{\mathbf{A}}_j \) 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 \( \hat{\mathbf{A}}_1 \). This is shown in the next theorem.

**Theorem 3.3.** Assume that the matrix \( A \) of (1.1) 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 \( Q_1U \) is less than \( k - 1 \). Moreover, the \((n-1) \times (n-1)\) trailing principal submatrix \( \hat{\mathbf{A}}_1 \) of \( A_1 = Q_1AQ_1^* \) has lower quasiseparable rank \( k \).

**Proof.** Observe that \( \hat{\mathbf{A}}_1 \) can be written as the sum of a Hermitian matrix and of a matrix of rank at most \( k \), namely, \( \hat{\mathbf{A}}_1 = \hat{Q}_1D\hat{Q}_1^* + \hat{\mathbf{U}}_1\hat{\mathbf{V}}_1^* \), where \( \hat{\mathbf{U}}_1, \hat{\mathbf{V}}_1 \) and \( \hat{Q}_1 \) are the matrices obtained by removing the first row of \( Q_1U \), \( Q_1V \), and \( Q_1 \), respectively. Define \( x = V^*e_1 \) so that \( x \neq 0 \) and \( Ae_1 = de_1 + Ux \). Since \( Q_1Ae_1 = \alpha_1e_1 + \beta_1e_2 \), we find that \( Q_1Ux = Q_1Ae_1 - de_1 \beta_1e_2 = (\alpha_1 - d_1)e_1 + \beta_1e_2 \), which implies \( \hat{U}_1x = \beta_1e_1 \). Thus, \( \hat{U}_1[2 : n - 1, \cdot] \) has rank at most \( k - 1 \). This matrix coincides with the matrix obtained by removing the first two rows of \( Q_1U \) so that the first part of the theorem is proven. Since \( \hat{U}_1[2 : n - 1, \cdot] \) has rank at most \( k - 1 \), then also the matrix \( \hat{U}_1[2 : n - 1, \cdot]\hat{\mathbf{V}}_1^* \) has rank at most \( k - 1 \). We can conclude that every submatrix in the strictly lower triangular part of \( \hat{\mathbf{A}}_1 \), given by the sum of a submatrix of \( \hat{U}_1[2 : n - 1, \cdot]\hat{\mathbf{V}}_1^* \) which has rank at most \( k - 1 \), and a submatrix in the lower triangular part of \( Q_1DQ_1^* \) which has rank at most \( 1 \) in view of Corollary 1 can have rank at most \( k \). This completes the proof. \( \square \)

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.

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 $O(n^2 k)$.

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 2.** A tuple $\mathcal{G} = (G_i)_{i \in I}$ on some ordered index set $(I, \leq)$ is said a sequence of Givens rotations. We also define $\prod_{i \in I} G_i$ the product in ascending order while $\prod_{i \in \text{rev}(I)} G_i$ denotes the product in descending order. The following operations on $\mathcal{G}$ are introduced:

- $Gv := \prod_{i \in \text{rev}(I)} G_i v$, for $v \in \mathbb{C}^n$;
- $G^* v := \prod_{i \in I} G_i^* v$ 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 $\mathcal{G} = (G_i)_{i \in I}$, $\mathcal{G}' = (G'_j)_{j \in J}$, we define the product $\mathcal{G} \mathcal{G}'$ to be the sequence

$$\mathcal{G} \mathcal{G}' := (E_i)_{i \in I \sqcup J}, \quad E_i = \begin{cases} G_i & \text{if } i \in I \\ G'_i & \text{if } i \in J \end{cases}$$

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, $\mathcal{G}A := \prod_{i \in \text{rev}(I)} G_i A$.

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 3.** 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 2 specialize in the following way:

- $Gv := G_{n-1} \ldots G_2 v$, for $v \in \mathbb{C}^n$;
- $G^* v := G_2^* \ldots G_{n-1}^* v$, for $v \in \mathbb{C}^n$;
- $\mathcal{G}[i : j] := (G_i, \ldots, G_j)$, for $2 \leq i < j \leq n - 1$, 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 $\mathcal{G}[i:]$ or $\mathcal{G}[:j]$ for $\mathcal{G}[i : n - 1]$ and $\mathcal{G}[2 : j]$, 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 $\square$ describes a Givens rotation $G_i$, applied to the pair $(i, i + 1)$ of consecutive rows.

$$\mathcal{G} = (G_2, \ldots, G_n) = \begin{bmatrix} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \end{bmatrix}, \quad \mathcal{G}v = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix}. $$
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

\[
(i_1, j_1) \leq_G (i_2, j_2) \iff j_1 > j_2 \text{ or } (j_1 = j_2 \text{ and } i_1 \leq i_2) .
\]

**Definition 4.** We say that \( \mathcal{G} = (G_{i,j})_{(i,j) \in I} \) is a \( k \)-sequence of Givens rotations if \( I = \{(i,j) \in \mathbb{N}^2 \mid i = 2, \ldots, n - 1, j = 1, \ldots, \min(i - 1, k)\} \) with the order induced by \( \leq_G \). With a slight abuse of notation we define the sequence \( \mathcal{G}[i_1 : i_2] := (G_{i,j})_{(i,j) \in I'} \) where \( I' = \{(i,j) \in \mathbb{N}^2 \mid i = i_1, \ldots, \min(i_2 + k, n - 1), j = \max(i, i_2 + 1), \ldots, \min(k, i - i_1 + 1)\}, \) \( 2 \leq i_1 < i_2 \leq n - 1 \) to be a slice of \( \mathcal{G} \) from \( i_1 \) to \( i_2 \), where the ordering in \( I' \) is induced by the ordering \( \leq_G \) 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

\[
\begin{array}{cccc}
\mathcal{G} = & & & \\
& & & \\
& & & \\
& & & \\
\end{array}
\]

(4.1)

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

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

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 \( \leq_G \) 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 (4.1).

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 \( G_{4,2}, G_{5,2}, G_{3,1}, G_{4,1} \).

\[
\begin{array}{cccc}
\mathcal{G} = & & & \\
& & & \\
& & & \\
& & & \\
\end{array}
\]

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

**Definition 5.** 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 $i = 1, \ldots, n - 1$ the subdiagonal elements of the $i$-th column of $A$ are equal to the last $n - i$ elements of $G[i + 1 : |w|]$, where we define

\[
    w_i := \begin{bmatrix} 0_i \\ W e_i \end{bmatrix} \quad \text{if } i < n - k,
\]
\[
    w_i := [(W e_i)[1 : n - i]] \quad \text{otherwise}
\]

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

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

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 $G$, a matrix $W \in \mathbb{C}^{k \times (n - 1)}$ and a diagonal matrix $D$ such that $A = \text{GV}(G, W, D)$.

• If $A = \text{GV}(G, W, D)$ for some $k$-sequence $G$, $W \in \mathbb{C}^{k \times (n - 1)}$ and $D$ diagonal, then $A$ is $j$-quasiseparable with $j \leq k$, i.e., the existence of the representation guarantees that $A$ 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 $G_i G_j - G_j G_i = 0$. 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 4.1.** Let $G$ 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 $\hat{G}$ and a Givens rotation $\hat{F}_{i - 1}$ acting on the rows $i - 1$ and $i$ such that

\[
    G F_i = \hat{F}_{i - 1} \hat{G}.
\]

Moreover, $\hat{G}$ differs from $G$ 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.

\[
\begin{array}{c}
\begin{array}{c}
\circ \circ \circ \\
\end{array}
\end{array} = \begin{array}{c}
\begin{array}{c}
\circ \circ \circ \\
\end{array}
\end{array}
\]

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

**Corollary 2.** Let $G$ 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 $\hat{G}$ and a Givens rotation $\hat{F}_{i - k}$ acting on the rows $i - k$ and $i - k + 1$ such that

\[
    G F_i = \hat{F}_{i - k} \hat{G},
\]

where $\hat{F}_{i - k} = I$ if $i - k \leq 0$. Moreover, $\hat{G}$ differs from $G$ 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 $i > k + 1$. 

12
We answer now to the following question: Given a $k$-sequence $G$ and a matrix $A$ of quasiseparable rank $k$, there exist appropriate matrices $W \in \mathbb{C}^{k \times (n-1)}$ and $D$ diagonal such that $A = GV(G, W, D)$.

**Lemma 4.2.** Let $A$ be a Hermitian matrix and $G$ a $k$-sequence of Givens rotations. Then $B = G^* A$ 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 $GV(G, W, D)$ for some $W \in \mathbb{C}^{k \times n}$ and $D$ real diagonal.

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

$$G^* \text{tril}(A,-1)e_i = G^*[i+1:]w_i = G^*[i][i+1:]G[i+1:]w_i = G^*[i]w_i.$$

We also have

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

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

$$G^* Ae_i = G^* \text{triu}(A)e_i + G^* \text{tril}(A,-1)e_i = G^*[i](\text{triu}(A)e_i + w_i).$$

Now observe that the rotations inside $G^*[i]$ only act on the first $i + k$ rows. This implies that, since both $w_i$ and $\text{triu}(A)e_i$ have all the components with index strictly bigger than $i + k$ equal to zero, the same must hold for $G^*[i](w_i + \text{triu}(A)e_i)$, and this completes the proof. The converse is also true. In fact, if $G^* A$ is lower banded with bandwidth $k$ we can build $W$ by setting $We_i = (G^*[i+1:]Ae_i)[i+1:i+k]$ and $D$ equal to the diagonal of $A$. Then the equation $A = GV(G, W, D)$ 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 $G$ spans $U \in \mathbb{C}^{n \times k}$ if there exists $Z \in \mathbb{C}^{k \times k}$ such that $G^* U = Z$. This definition is motivated by the following

**Lemma 4.3.** If $G$ 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 $A_1 = UV^* + GV(G, W, D)$ is lower $k$-quasiseparable and $A_2 = t(UV^*) + GV(G, W, D)$ is $k$-quasiseparable. In particular, both $G^* A_1$ and $G^* A_2$ are lower banded with bandwidth $k$.

**Proof.** For the first part of the Lemma it suffices to observe that $G^* A_1$ is lower banded with bandwidth $k$. This follows directly by noting that $A = GV(G, W, D) + UV^*$. Since $G^* UV^* = [Z^*] V^*$ and $G^* GV(G, W, D)$ is lower banded by Lemma 4.2 we conclude that also $G^* A_1$ 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 $G^* A_2$ is lower banded we can write

$$G^* A_2 = G^*(A_1 - \text{triu}(UV^*) + \text{triu}(VU^*, 1)) = G^* A_1 + G^* R,$$
where $R$ is upper triangular. Since $G^*$, represented as a matrix, is the product of $k$ upper Hessenberg matrices, it is lower banded with bandwidth $k$. This implies that also $G^*R$ is lower banded with bandwidth $k$ and so the same must hold also for $G^*A_2$.

**Remark 1.** The above Lemma shows how the Givens rotations in a GV representation of a matrix in $QSH_k$ 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. More precisely, in order to implement in terms of algorithms the constructive proofs given in Section 2, we need to explain how to efficiently perform the following tasks assuming we are given GV representations of $M = D + t(UV^*) = GV(G, W, D) \in QSH_k$ and of $S = D_S + t(W^*) \in QSH_1$, where $U, V \in \mathbb{C}^{n \times k}$, $G$ 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 1.3, we find that $G^*S$ is lower banded with bandwidth $k$ and so by applying Lemma 4.2 there exists $W_S \in \mathbb{C}^{n \times k}$ and $D_S$ real diagonal such that $S = GV(G, W_S, D_S)$ is a GV representation of $S$. Given an algorithm for the computation of $W_S$ it is possible to represent $M + S$ as $GV(G, W_S, W^* + D + D_S)$.

We need to investigate how to actually compute the matrix $W_S$ assuming we are given a GV representation $S = GV(F, z, D_S)$ of $S$. Recall that the $i$-th column of $W_S$ can be extracted from the components of the vector $G^*[i + 1:]Me_i$, as explained in Lemma 1.2. We compute the whole matrix $W_S$ at cost $O(nk)$ by following this procedure:

- Compute the last column of $W_S$ by using Lemma 1.2. This is almost cost-free since no rotations are involved, and the only significant element of $W_S \epsilon_{n-1}$ is equal to $z_{n-1}$.
- Compute $W_S \epsilon_{i+1}$, starting from $W_S \epsilon_{i+1}$; this vector can be computed by using some elements in $G^*[i+1:]Me_i$. In fact, since we are in the $1$-quasiseparable case then $z_i = z_{i+1}$.

So we have

$$G^*[i + 1:]Me_i = G^*[i + 1:]F[i + 1:]z_i = G^*[i + 1][i + 2:]F[i + 2:]z_i = z_iG^*[i + 2:]F[i + 2:]z_i (\alpha e_i + \beta e_{i+1}) = \frac{z_i}{z_{i+1}} G^*[i + 1] \left( \frac{\beta}{z_{i+1}} G^*[i + 2:]Me_{i+1} + \alpha e_i \right).$$

Since $G^*[i + 2:]Me_{i+1}$ has already been computed, we obtain $W_S \epsilon_{i+1}$ from $W_S \epsilon_{i+1}$ at the cost of $O(k)$ rotations. Some care needs to be taken because in the above formula it may happen that $z_{i+1}$ is zero. This issue can be solved by replacing $Me_i$ with $F[i + 1:]e_i$, and by multiplying the computed $W_S \epsilon_i$ by $z_i$ after the computation. This way, we can also avoid the possible cancellation in cases where the $z_i$ are unbalanced.

The above procedure provides a fast method for computing a $k$-quasiseparable representation of $S$.

We investigate now the problem of computing the residual matrix given by Theorem 2.4 and Corollary 1 using the GV representation of $M$ and $S$. We rephrase the proof of Theorem 2.4 in terms of GV representations.

Analyzing the proof of Theorem 2.4 and its corollaries, we can observe that the algorithm can be easily constructed if we are able to compute the residual $R_i = t(F_i UV^*) - F_i t(UV^*) F_i^*$ 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 all the cases by just adding some care in the process.
Assume we are given \(D, U, V, W\) and \(G\) such that \(M = D + t(UV^*) = GV(G, W, D)\). Observe that we can compute an updated \(\hat{G}\) such that \(\hat{G}\) spans \(F_i U\). In fact, we know that \(G^* F_i^* F_i U\) is of the form \([\begin{matrix} x \\ 0 \end{matrix}]\) where \(x\) is an appropriate \(k \times k\) block. The rotation \(F_i^*\) can be passed through the rotations inside \(G^*\) (by properly updating them using the turnover operation) obtaining \(\hat{F}_{i+k}^* G^* = G^* F_i^*\). Then, by \(\hat{F}_{i+k}^* G^* F_i U = \begin{bmatrix} x \\ 0 \end{bmatrix}\), we can conclude that also \(\hat{G}^* F_i U = \hat{F}_{i+k} \begin{bmatrix} x \\ 0 \end{bmatrix}\) since \(\hat{F}_{i+k}\) is operating on the null rows.

Moreover, we can check that \(GV(\hat{G}, W, D)\) 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 \(\hat{G}[j + 1 :] = G[j + 1 :]\) 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 \(\text{tril}(t(F_i UV^* F_i^*), -1)e_j = \text{tril}(F_i UV, -1)e_j\). Recall that, by definition of \(GV\) representation, we have

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

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

\[
\text{tril}(F_i UV^*,-1)e_j = F_i G[j + 1 :]w_i = \hat{G}[j + 1 :]\hat{F}_{i+k} w_j = \hat{G}[j + 1 :]w_j
\]

where the last two equalities follow from the definition of \(\hat{G}\) and from the fact that the components of \(w_j\) with index bigger than \(j + k\) are zero. Thus we have that \(GV(\hat{G}, W, D)\) provides a good representation of the lower part of the \(j\)-th column of \(t(F_i UV^* F_i^*)\), 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 (4.2) represents the first case, where the rotation \(F_i\) does not intersect the indices of the rotations in \(G[j + 1 :]\), and Equation (4.3) the latter case, where an update of the rotations is necessary.

\[
\begin{bmatrix}
  0 \\
  ⋮ \\
  0 \\
  * \\
  * \\
  0 \\
  0 \\
\end{bmatrix}
\begin{bmatrix}
  0 \\
  ⋮ \\
  0 \\
\end{bmatrix} =
\begin{bmatrix}
  0 \\
  ⋮ \\
  0 \\
  * \\
  * \\
  0 \\
  0 \\n\end{bmatrix}
\]
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):

\[
M \begin{bmatrix} e_i & e_{i+1} \end{bmatrix} = G[i + 2 :] \left( \begin{bmatrix} d_i & 0 \\ w_{1,i} & 0 \\ w_{2,i} & 0 \\ w_{3,i} & 0 \end{bmatrix} + \begin{bmatrix} 0 & \times \\ 0 & d_{i+1} \\ 0 & w_{1,i+1} \\ 0 & w_{2,i+1} \end{bmatrix} \right) \cdot
\]

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

\[
F_iM F_i^* \begin{bmatrix} e_i & e_{i+1} \end{bmatrix} = G[i + 2 :] \left( \begin{bmatrix} d_i & 0 \\ w_{1,i} & 0 \\ w_{2,i} & 0 \\ w_{3,i} & 0 \end{bmatrix} + \begin{bmatrix} 0 & \times \\ 0 & d_{i+1} \\ 0 & w_{1,i+1} \\ 0 & w_{2,i+1} \end{bmatrix} \right) \cdot
\]

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

\[
\begin{bmatrix} \hat{c} & \hat{d}_i & 0 \\ \hat{w}_{1,i} & 0 & \times \\ \hat{w}_{2,i} & 0 & d_{i+1} \\ \hat{w}_{3,i} & 0 & w_{1,i+1} \\ 0 & w_{2,i+1} & 0 \end{bmatrix} + \begin{bmatrix} \hat{d}_i & 0 \\ \hat{w}_{1,i} & 0 & \times \\ \hat{w}_{2,i} & 0 & d_{i+1} \\ \hat{w}_{3,i} & 0 & w_{1,i+1} \\ 0 & w_{2,i+1} & 0 \end{bmatrix}
\]

(4.4)
Let $C$ be the left matrix in \[4.4\]. Then the elements $\hat{w}_{j,i+1}$ must coincide with the vector $C[3,2]$, the diagonal elements $\hat{d}_i$ and $\hat{d}_{i+1}$ are determined by the diagonal of the top $2 \times 2$ block of $C$. It remains to determine the elements $\hat{w}_{j,i}$ and the value $\alpha$. To find them we can multiply on the left by the inverses of the rotations in $\hat{C}[i]$. We get the equation

\[
\hat{C}^{-1} \left( \begin{array}{c} \alpha e_1 \\
0 \\
0 \\
0 \\
0 \\
\end{array} \right) = \left( \begin{array}{c} \hat{d}_i \\
\hat{w}_1,i \\
\hat{w}_2,i \\
\hat{w}_3,i \\
0 \\
\end{array} \right).
\]

We can choose $\alpha$ 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 $\hat{w}_{j,i}$ by back substitution.

The algorithm presented above allows us to find the residual matrix $R$ and a new $k$-quasiseparable matrix $\hat{M}$ with the desired rotations in its representations such that $\hat{M} + R = F^*_1MF^*_2$. This is what we desire in order to implement the algorithm. The reader may wonder if $\hat{M}$ is exactly the matrix $t(F^*_1UVF^*_2)$. This is actually true by assuming some hypothesis about the irreducibility of the matrices involved and that $i$ is not bigger than $n - k - 1$, essentially due to the fact that $F^*_1MF^*_2$ is forced to be equal to $t(F^*_1UVF^*_2)$ in all the elements but the $2 \times 2$ submatrix of indices $(i, i + 1)$. We omit the analysis of this property since, in the end, it is not very relevant for the algorithm. The important result is just that we have found a matrix with the correct Givens rotations such that the sum of the lower parts is contained in the same span. This last claim clearly holds in this case.

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 2.4, this form is maintained by the trailing principal submatrices $\hat{A}_j \in \mathbb{C}^{(n-j+1) \times (n-j+1)}$ of the matrices $A_j$, generated at each step $j$ of the algorithm, that is, $\hat{A}_j = t(U_jW_j^*) + S_j + U_jV_j^*$. 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 2.2

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 $G$ such that $G v = v_1 e_1 + \alpha e_2$ for some $\alpha$.
- **(R_M, M) ← conjugateAndTruncate(M, G)** takes as input a quasiseparable Hermitian matrix $M \in \mathbb{QSH}_k$ and a sequence of Givens rotations $G$. Then it computes a quasiseparable representation for $GMG^* - R_M$ where $R_M$ is a matrix in $\mathbb{QSH}_1$. It returns an updated representation of $M$ and the residual matrix $R_M$.
- $S ← R_M + R_S$ computes the sum of the matrices $R_M, R_S \in \mathbb{QSH}_1$. 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 $M$. 17
Algorithm 2 High level reduction process

1: $A_1 \leftarrow D + UV^*$
2: $M \leftarrow 0$
3: $S \leftarrow 0$
4: $s \leftarrow \text{zeros}(1, n - 1)$
5: $d \leftarrow \text{zeros}(1, n)$
6: for $i = 1, \ldots, n - 2 - k$ do
7: $G \leftarrow \text{cleanColumn}(A_i[:, 1])$
8: $d[i] \leftarrow (G A_i[:, 1])[1]$
9: $s[i] \leftarrow (G A_i[:, 1])[2]$
10: $U \leftarrow (G \cdot U)[2 : n, :]$
11: $V \leftarrow (G \cdot V)[2 : n, :]$
12: $(R_M, M) \leftarrow \text{conjugateAndTruncate}(M, G)$
13: $(R_S, S) \leftarrow \text{conjugateAndTruncate}(S, G)$
14: $M \leftarrow M + S$
15: $S \leftarrow R_M + R_S$
16: end for
17: $(d[n - 1 - k : n], s[n - 1 - k : n - 1], U, V) \leftarrow \text{reduceTrailingBlock}(A_{n-1-k})$

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 $G^*U = [Z_0]$. 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 $G^*U$ in the desired form and correct the moduli of the sine and cosine inside $G$ 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:

- 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 6.1, we have reported, in log scale, the timings for some experiments with $n = 100i$ for $i = 1, \ldots, 10$. In Figure 6.2, 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.
Figure 6.1. 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 \( \gamma n^2 \) for an appropriate \( \gamma \). It is evident the quadratic behavior of the time.

| 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 6.2. CPU time, in seconds, for the Hessenberg reduction of a \( 400 \times 400 \) diagonal plus rank \( k \) matrix.

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

Looking at the results in Figure 6.2, 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 6.3 and Figure 6.4 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.

The matrices in these examples have been obtained by using the \texttt{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 \( A = D + UV^* \).

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 (1.1) 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 $O(dk^2)$ so that the computation at $n = dk$ different points has the cost $O(dk^2)$. 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 $O(dk^2)$ 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 $(xI - H)v = \ldots$
αe_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 \( \alpha \) 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 \( \alpha \) at the same cost. On the other hand, by the Cramer rule we find that
\[
1 = v_n = \frac{\alpha \prod_{i=2}^{n} h_{i, i-1}}{\det(xI - H)}
\]
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 \( p(x) = \det(xI - H) \).

REFERENCES

[1] Dario A. Bini and Vanni Noferini. Solving polynomial eigenvalue problems by means of the ehrlich-aberth method. *Linear Algebra Appl.*, 439:1130–1149, 2013.
[2] Dario A Bini and Leonardo Robol. On a class of matrix pencils equivalent to a given matrix polynomial. *arXiv preprint arXiv:1406.1025*, 2014.
[3] Dario A. Bini and Leonardo Robol. Solving secular and polynomial equations: a multiprecision algorithm. *J. Comput. Appl. Math.*, 272:270–292, 2014.
[4] Yuli Eidelman, Israel Gohberg, and Luca Gemignani. On the fast reduction of a quasiseparable matrix to hessenberg and tridiagonal forms. *Linear algebra and its applications*, 420(1):86–101, 2007.
[5] Yuli Eidelman, Israel Gohberg, and Iulian Haimovici. *Separable type representations of matrices and fast algorithms. Vol. 1*, volume 234 of *Operator Theory: Advances and Applications*. Birkhäuser/Springer Basel, Basel, 2014. Basics. Completion problems. Multiplication and inversion algorithms.
[6] Yuli Eidelman, Israel Gohberg, and Iulian Haimovici. *Separable type representations of matrices and fast algorithms. Vol. 2*, volume 235 of *Operator Theory: Advances and Applications*. Birkhäuser/Springer Basel AG, Basel, 2014. Eigenvalue method.
[7] Yuli Eidelman, Israel Gohberg, and Vadim Olshevsky. The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order. *Linear Algebra Appl.*, 404:305–324, 2005.
[8] Katrijn Frederix, Steven Delvaux, and Marc Van Barel. An algorithm for computing the eigenvalues of block companion matrices. *Numerical Algorithms*, 62(2):261–287, 2013.
[9] Brian T. Smith. Error bounds for zeros of a polynomial based upon Gerschgorin’s theorems. *J. Assoc. Comput. Mach.*, 17:661–674, 1970.
[10] Ellen Van Camp. *Diagonal-plus-semiseparable matrices and their use in numerical linear algebra*. PhD thesis, PhD thesis, Katholieke Universiteit Leuven, 2005.
[11] Raf Vandebril, Marc Van Barel, and Nicola Mastronardi. An implicit QR algorithm for symmetric semiseparable matrices. *Numer. Linear Algebra Appl.*, 12(7):625–658, 2005.
[12] Raf Vandebril, Marc Van Barel, and Nicola Mastronardi. *Matrix computations and semiseparable matrices. Eigenvalue and singular value methods*, volume 2. Johns Hopkins University Press, Baltimore, MD, 2008.
[13] Raf Vandebril, Marc Van Barel, and Nicola Mastronardi. *Matrix computations and semiseparable matrices. Linear systems*, volume 1. Johns Hopkins University Press, Baltimore, MD, 2008.
[14] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, 1965.