Numer. Math. (2005)
Digital Object Identifier (DOI) 10.1007/s00211-005-0595-4

Numerische Mathematik

Fast and stable QR eigenvalue algorithms for generalized companion matrices and secular equations ⋆

Dario A. Bini[1], **Luca Gemignani**[1], **Victor Y. Pan**[2]

[1] Dipartimento di Matematica, Università di Pisa, Via F. Buonarroti 2, 56127 Pisa, Italy; e-mail: {gemignan, bini@dm.unipi.it}dm.unipi.it.
[2] Department of Mathematics and Computer Science, Lehman College of CUNY, Bronx, NY 10468, USA; e-mail: vpan@lehman.cuny.edu.

Received May 31, 2005 / Revised version received December 29, 2004
Published online xxx 2005,

Summary. We introduce a class $\mathcal{C}_n$ of $n \times n$ structured matrices which includes three well-known classes of generalized companion matrices: tridiagonal plus rank-one matrices (comrade matrices), diagonal plus rank-one matrices and arrowhead matrices. Relying on the structure properties of $\mathcal{C}_n$, we show that if $A \in \mathcal{C}_n$ then $A' = RQ \in \mathcal{C}_n$, where $A = QR$ is the QR decomposition of A. This allows one to implement the QR iteration for computing the eigenvalues and the eigenvectors of any $A \in \mathcal{C}_n$ with $O(n)$ arithmetic operations per iteration and with $O(n)$ memory storage. This iteration, applied to generalized companion matrices, provides new $O(n^2)$ flops algorithms for computing polynomial zeros and for solving the associated (rational) secular equations. Numerical experiments confirm the effectiveness and the robustness of our approach.

Mathematics Subject Classification (2000): 65F15, 65H17

⋆ The results of this paper were presented at the Workshop on Nonlinear Approximations in Numerical Analysis, June 22 – 25, 2003, Moscow, Russia, at the Workshop on Operator Theory and Applications (IWOTA), June 24 – 27, 2003, Cagliari, Italy, at the Workshop on Numerical Linear Algebra at Universidad Carlos III in Leganes, June 16 – 17, 2003, Leganes, Spain, at the SIAM Conference on Applied Linear Algebra, July 15 – 19, 2003, Williamsburg, VA and in the Technical Report [8].

This work was partially supported by MIUR, grant number 2002014121, and by GNCS-INDAM.
This work was supported by NSF Grant CCR 9732206 and PSC CUNY Awards 66406-0033 and 65393-0034.

Correspondence to: L. Gemignani

2 1 1 0 5 9 5 B | Jour. No | Ms. No.
Dispatch: 17/3/2005 | Total pages: 36 | Disk Received ☑ | Disk Used ☑
Journal: Numer. Math. | Not Used ☐ | Corrupted ☐ | Mismatch ☐

1 Introduction

Matrix methods based on the QR algorithm are customary for polynomial root-finding: they are proposed to the users of MATLAB[1]. These root-finders enjoy the robustness and the rapid convergence properties of the QR algorithm (see [48,38,3,59,60] for a general account of these properties) but also inherit its substantial cost of the order of n^2 arithmetic operations per iteration and quadratic memory space for a polynomial of degree n. Our goal is to present a QR-based root-finder for some specific classes of polynomial and rational equations which runs in linear time per iteration and uses linear memory space. The algorithm is not just a variant but rather a fast adaptation of the classical QR iteration which exploits the structure of the associated matrix problems. Therefore, it remains robust and converges as fast as the customary QR algorithm.

More specifically, we present a QR-based algorithm for the computation of the eigenvalues of some classes of $n \times n$ generalized companion matrices in $O(n)$ arithmetic operations per iteration and with $O(n)$ memory storage. As a by product, this algorithm computes all the zeros of the n-degree characteristic polynomial of a generalized companion matrix as well as the associated secular equation in $O(n^2)$ flops[2]. Once the eigenvalues are available, the whole set of eigenvectors can in principle be computed efficiently by means of the inverse power method at the cost of $O(n)$ flops per iteration.

The generalized companion matrices which we consider include

- arrowhead matrices, that is, matrices which have null elements everywhere except for the elements on the diagonal, in the first row and in the first column [1].
- comrade matrices [4], that is, matrices which are real symmetric tridiagonal except in the last column where the elements can take any value;
- diagonal plus rank one matrices.

The key idea of our approach is to find the class $\mathcal{C}_n$ of structured matrices defined by the minimum number of parameters, which includes the above generalized companion matrices and which is closed under the QR iteration. That is, $\mathcal{C}_n$ is such that, if $A \in \mathcal{C}_n$ and $A - \sigma I = QR$ is the QR factorization of $A - \sigma I$ then $A' = RQ + \sigma I \in \mathcal{C}_n$ for any complex σ. In this way, the sequence $\{A_s\}$ generated by the QR iteration

$$\begin{aligned} &A_0 = A \\ &A_s - \sigma_s I = Q_s R_s \\ &A_{s+1} := R_s Q_s + \sigma_s I \end{aligned} \tag{1.1}$$

[1] MATLAB is a registered trademark of The Mathworks, Inc..

[2] A *flop* is a floating point operation $x \circ y$, where x and y are floating point numbers and $\circ$ denotes one of $+, -, \times, \div$.

for $s = 0, 1, \ldots$ and for any choice of σ_s, is such that $A_s \in \mathcal{C}_n$ for any $s \geq 0$. Here, Q_s is unitary and R_s is upper triangular for $s \geq 0$. We define $\mathcal{C}_n$ in Section 1.1.

As pointed out in [1,2], the QR iteration (1.1) does not preserve the structure of generalized companion matrices. For instance, if A_0 is an arrowhead matrix then A_1 is not generally an arrowhead matrix. However, as we show in Section 3, the QR iteration preserves the structure of the matrices in $\mathcal{C}_n$.

1.1 The class $\mathcal{C}_n$

A matrix $A = (a_{i,j}) \in \mathbb{C}^{n\times n}$ belongs to $\mathcal{C}_n$ if there exist real numbers $d_1, \ldots, d_n$, complex numbers $t_2, \ldots, t_{n-1}$, and four complex vectors $\boldsymbol{u} = [u_1, \ldots, u_n]^T, \boldsymbol{v} = [v_1, \ldots, v_n]^T, \boldsymbol{w} = [w_1, \ldots, w_n]^T$ and $\boldsymbol{z} = [z_1, \ldots, z_n]^T$ in $\mathbb{C}^n$ such that

$$(1.2)\quad \begin{cases} a_{i,i} = d_i + z_i\overline{w_i}, & 1 \leq i \leq n; \\ a_{i,j} = u_i t^{\times}_{i,j}\overline{v_j}, & 1 \leq j < i,\ 2 \leq i \leq n; \\ a_{i,j} = \overline{u_j} t^{\times}_{j,i} v_i + z_i\overline{w_j} - \overline{z_j} w_i, & 1 \leq i < j,\ 2 \leq j \leq n, \end{cases}$$

where $t^{\times}_{i,j} = t_{i-1} \ldots t_{j+1}$ for $i - 1 \geq j + 1$ and, otherwise, $t^{\times}_{i,i-1} = 1$.

It turns out that $\mathcal{C}_n$ is a subclass of some well known classes of matrices such as the matrices with low Hankel rank of [20,19], the quasiseparable matrices introduced and studied in [23,25], the recursively semi-separable and sequentially semi-separable matrices described in [15,16] and the weakly semiseparable matrices defined in [50]. These classes arise in the numerical solution of discretized rational approximation problems and integral equations with structured kernels (see, e.g., [28,29,14,22,23,25,53]) and, essentially, contain matrices which have low rank on their off-diagonal blocks (see also [56] for comparisons among different definitions). In particular, a matrix $B \in \mathbb{C}^{n\times n}$ is called quasiseparable of order (r, s) if $\operatorname{rank} B[k+1\colon n, 1\colon k] \leq r$ and $\operatorname{rank} B[1\colon k, k+1\colon n] \leq s$ for $k = 1, \ldots, n-1$, where we adopt the MATLAB notation $B[i\colon j, k\colon l]$ for the submatrix of B with entries having row and column indices in the ranges i through j and k through l, respectively. A matrix $A \in \mathcal{C}_n$ is therefore a quasiseparable matrix of order (1, 3) with a suitable rank structure in the upper triangular corner.

Appropriate choices for the elements defining the structured representation of A turn the class $\mathcal{C}_n$ into the cited classes of generalized companion matrices.

- For $\boldsymbol{z} = \boldsymbol{u}$, $\boldsymbol{w} = \boldsymbol{v}$ and $t_i = 1$, $i = 2, \ldots, n-1$, the class $\mathcal{C}_n$ contains the diagonal plus rank-one matrices of the form $A = D + \boldsymbol{u}\boldsymbol{v}^H$ with $D = \operatorname{diag}[d_1, \ldots, d_n] \in \mathbb{R}^{n\times n}$.
- Under the assumptions $\boldsymbol{w} = w\boldsymbol{e}_n$ and $t_i = 0, i = 2, \ldots, n-1$, A reduces to a Hermitian tridiagonal matrix plus the rank-one correction $w\boldsymbol{z}\boldsymbol{e}_n^T$.

– If $\boldsymbol{w} = \boldsymbol{z}$ and $t_i \neq 0$ for $i = 2, \ldots, n-1$, then A is a Hermitian diagonal-plus-semiseparable (*dpss*) matrix [41].
– Finally, for $v_2 = \ldots = v_n = 0$ and $z_2 = \ldots = z_n = 0$, A turns into an arrowhead matrix.

The property of rank invariance under the QR iteration was first observed and communicated to us by D. Fasino in the case of Hermitian positive definite dpss matrices, where the diagonal-plus-semiseparable structure is maintained during the QR process. Fast implementations of the QR iteration for real symmetric dpss matrices are also described in [55,57,58] together with several applications to computing the SVD and rank-revealing factorizations of symmetric matrices.

1.2 The QR iteration

Once we have proved that all the matrices A_k, $k \geq 0$ generated by (1.1) with $A_0 \in \mathcal{C}_n$ belong to $\mathcal{C}_n$, our next step is to relate the $6n-2$ parameters which define the generic matrix $A_s \in \mathcal{C}_n$ by means of (1.2) with the $6n-2$ parameters which define the matrix A_{s+1}. This is done in Section 4 where we design an algorithm that performs this computation in about $120n$ ops.

In principle, one may perform the QR iteration in linear time per iteration by extending the QR factorization algorithm in [24]. Indeed, since A_s is quasiseparable of order $(1, 3)$ and one iteration of the QR algorithm applied to A_s generates another quasiseparable matrix A_{s+1} of order $(1, 3)$, the algorithm given in [24] can be used for the computation of a QR factorization of each matrix A_s generated by the QR process. Due to the very special rank structure of the upper triangular part of A_s, however, one finds that the quasiseparable representations of Q_s and R_s also satisfy additional requirements which should be forcefully imposed during the computation. Moreover, since R_s is quasiseparable of order $(0, 4)$, the algorithm of [24] works with a parametrization which is both mathematically and numerically redundant.

To circumvent these difficulties, we propose a modification of the QR factorization scheme of [24], where the upper triangular factor R_s is never explicitly formed but implicitly represented as the sum of products of triangular quasiseparable matrices. Then we show that the structural representation (1.2) of A_{s+1} can be recovered from the structural representation of A_s and from certain quantities generated step-by-step in the process of multiplying R_s by Q_s, where the matrices Q_s are kept in factored form as products of elementary Givens rotations. This makes our design more involved technically but decreases the number of parameters required to accurately determine A_{s+1} in the presence of numerical errors and consequently decreases the number of flops in our algorithm.

1.3 Motivation and related work

Our interest in structured eigenproblems originates from our search for numerically reliable methods for approximating all roots of a high degree polynomial $p(z)$ [6]. One may reduce this problem to the computation of the eigenvalues of a matrix called generalized companion matrix for $p(z)$ whose characteristic polynomial has the same zeros as $p(z)$. The concept of *secular equation* also provides the means to convert polynomial root-finding problem into matrix eigenvalue computations. The name is due to Golub [37] who considered the solution of secular equations arising in some modified eigenvalue problems.

The simplest approach is to consider the associated Frobenius (companion) matrix (which is an upper Hessenberg matrix) having $p(z)$ as its characteristic polynomial. In particular, MATLAB employs the QR algorithm for the computation of the eigenvalues of a Frobenius matrix to approximate the zeros of a polynomial given by its coefficients. However, as Cleve Moler has pointed out [46], this method may not be the best possible because "it uses order n^2 storage and order n^3 time. An algorithm designed specifically for polynomial roots might use order n storage and n^2 time." Computational advances along this direction have been first obtained in [52,51].

It is worth pointing out that Frobenius matrices are a special instance of the more general set of rank-one perturbations of unitary Hessenberg matrices (fellow matrices) which appear in the context of root-finding for linear combinations of Szegö polynomials [12]. A fast $O(n^2)$ QR-algorithm for eigenvalue computation of unitary Hessenberg matrices has been first designed by Gragg [40] by exploiting their representation as a product of Givens rotation matrices. Since fellow matrices do not inherit such a factorization, available implementations of the fast QR-algorithm from [40] for unitary Hessenberg matrices, however, are not able to use the structure of these matrices and typically require $O(n^3)$ flops and $O(n^2)$ storage locations.

Other generalized companion matrices have been also proposed for devising new polynomial root-finders or for rephrasing the known functional iterations into a matrix setting. In [4] generalized companion (comrade) matrices of the form $A = T + \boldsymbol{u}\boldsymbol{v}^H$, where $T \in \mathbb{C}^{n\times n}$ is a Hermitian tridiagonal matrix, are considered for dealing with polynomials expressed with respect to an orthogonal polynomial basis satisfying a three-terms recursion. Matrices of the form $A = D + \boldsymbol{u}\boldsymbol{v}^H$, where D is a diagonal matrix, are introduced in [26,13] and used in [6] to provide a matrix formulation of the Weierstrass (Durand-Kerner) method and in [31] for the design of an $O(n^3)$ eigenvalue algorithm for approximating polynomial roots. Furthermore, a root-finding method is devised in [42] which is based on matrix computations applied to a generalized companion matrix of arrowhead form first studied in [30].

For two subclasses of generalized companion matrices, that is, arrowhead matrices and rank-one perturbations of diagonal matrices, it is shown in [37] that their eigenvalues are the roots of a secular equation of the form

$$\alpha z + \beta + \sum_{j=1}^{n} \frac{p_i}{z - z_i} = 0.$$

Conversely, for such an equation we may easily compute a generalized companion matrix of these classes; therefore our algorithm approximates the roots of this equation. Secular equations of this form play an important role, e.g., in updating the singular value decomposition (SVD) of matrices [10] and in divide-and-conquer algorithms for computing the eigenvalues of Hermitian tridiagonal matrices [17,21]. Related secular equations are also used for solving least squares type problems [33,34,39], in invariant subspace computations [32] and in the "escalator method" for computing the eigenvalues of a matrix [27]. Numerical methods for solving secular equations based on functional iterations are proposed, e.g., in [11,47,49,44,45,9].

The TR version [8] of this paper extended our previous work in [6] on polynomial root-finding by means of matrix methods. A large part of our techniques on structured matrices is related to [24] although neither root-finding applications nor the properties of the QR iteration are considered in [24].

1.4 Paper organization

The paper is organized as follows. In Section 2 we set up notations. In Section 3 we prove the invariance of the structure (1.2) under the QR iteration. In Section 4 we develop a fast algorithm for the QR step applied to a matrix $A \in \mathcal{C}_n$ with the cost of about $120n$ ops. In Section 5 we summarize our QR-based algorithm and present the results of extensive numerical experiments. Finally, conclusion and discussion are the subjects of Section 6.

2 Notation

We use capital letters for matrices and lower boldface letters for vectors. We denote by $A = (a_{i,j}) = \text{triu}(B, p)$ the upper triangular portion of B formed by the elements on and above the p-th diagonal of B, that is, $a_{i,j} = b_{i,j}$ for $j - i \geq p$, and $a_{i,j} = 0$ elsewhere. Analogously, the $n \times n$ matrix $A = \text{tril}(B, p)$ is formed by the elements on and below the p-th diagonal of B, that is, $a_{i,j} = b_{i,j}$ for $j - i \leq p$, and $a_{i,j} = 0$ elsewhere.

Given two sequences $\{a_i\}_{i=1}^n$ and $\{b_i\}_{i=1}^m$, then $\{a_i\}_{i=1}^n \odot \{b_i\}_{i=1}^m$ is the sequence obtained by concatenating them, i.e.,

$$\{a_i\}_{i=1}^n \odot \{b_i\}_{i=1}^m = \{a_1, \ldots, a_n, b_1, \ldots, b_m\}.$$

Let $A = (a_{i,j}) \in \mathbb{C}^{n\times n}$ be a matrix of the form given in (1.2). One has

$$\text{(2.1)}\quad \text{tril}(A,-1) = \begin{bmatrix} 0 & \dots & \dots & \dots & 0 \\ u_2\overline{v_1} & 0 & \ddots & \ddots & \vdots \\ u_3t_2\overline{v_1} & u_3\overline{v_2} & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ u_nt_{n-1}\dots t_2\overline{v_1} & u_nt_{n-1}\dots t_3\overline{v_2} & \dots & u_n\overline{v_{n-1}} & 0 \end{bmatrix},$$

where $\overline{v_j}$ denotes the complex conjugate of v_j. Given the elements $u_2, \dots, u_n$, $t_2, \dots, t_{n-1}$ and $\overline{v_1}, \dots, \overline{v_{n-1}}$, we denote by $L(\{u_i\}_{i=2}^n, \{\overline{v_i}\}_{i=1}^{n-1}, \{t_i\}_{i=2}^{n-1})$ the lower triangular matrix on the right hand side of (2.1). Moreover, the matrix

$$R(\{\overline{u_i}\}_{i=2}^n, \{v_i\}_{i=1}^{n-1}, \{\overline{t_i}\}_{i=2}^{n-1}) = (L(\{u_i\}_{i=2}^n, \{\overline{v_i}\}_{i=1}^{n-1}, \{t_i\}_{i=2}^{n-1}))^H,$$

is the upper triangular matrix of parameters $\overline{u_2}, \dots, \overline{u_n}$, $v_1, \dots, v_{n-1}$ and $\overline{t_2}, \dots, \overline{t_{n-1}}$.

Remark 2.1. A Hermitian matrix A satisfying (2.1) is characterized by the property of having only rank-one submatrices in its strictly lower and upper triangular parts. Historically, matrices with this property appeared for the first time in the study of the structure of the inverses of banded matrices as a part of a celebrated theorem due to Gantmacher and Krein [35]. More precisely, the theorem implied that if B is an invertible Hermitian irreducible tridiagonal matrix, then $A = B^{-1}$ satisfies (2.1) with $t_2 = \dots = t_{n-1} = 1$, that is, the matrix B^{-1} shares its strictly lower triangular part with a rank-one matrix. However, if B is assumed to be reducible, this result is not yet true. Taking $B = J_2 \oplus J_2$, where $J_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, it is immediately seen that there exist no $\boldsymbol{u}$ and $\boldsymbol{v}$ such that $\text{tril}(B^{-1}, -1) = \text{tril}(\boldsymbol{u}\boldsymbol{v}^H, -1)$, but nonetheless, $\text{tril}(B^{-1}, -1)$ still satisfies (2.1) for suitable parameters u_i, v_i and t_i. This means that the structure (2.1) generalizes the rank-one structure by providing a more robust representation for the cases where $\boldsymbol{u}$ and $\boldsymbol{v}$ have tiny elements.

We denote by $R'(\cdot)$ the submatrix of $R(\cdot)$ obtained after deletion of its first column and last row. Analogously, $L'(\cdot)$ is the submatrix of $L(\cdot)$ obtained after deletion of its first row and last column. Observe that $L(\{u_i\}_{i=2}^n, \{\overline{v_i}\}_{i=1}^{n-1}, 0)$ is a lower bidiagonal matrix with zero diagonal entries and subdiagonal entries equal to $\eta_i = u_i\overline{v_{i-1}}$, $2 \le i \le n$. Such a matrix is denoted by $\text{Subdiag}(\{\eta_i\}_{i=2}^n)$.

Let $A \in \mathcal{C}_n$ be defined by (1.2) and denote $\boldsymbol{x}_i = [z_i, w_i]$ and $\boldsymbol{y}_i = [w_i, -z_i]$ for $i = 1, \dots, n$. It is easily verified that

$$\text{triu}(A,1) - R(\{\overline{u_i}\}_{i=2}^{n}, \{v_i\}_{i=1}^{n-1}, \{\overline{t_i}\}_{i=2}^{n-1}) = \begin{bmatrix} 0 & \boldsymbol{x}_1\boldsymbol{y}_2^H & \dots & \boldsymbol{x}_1\boldsymbol{y}_n^H \\ & \ddots & \ddots & \vdots \\ & & \ddots & \boldsymbol{x}_{n-1}\boldsymbol{y}_n^H \\ & & & 0 \end{bmatrix}. \tag{2.2}$$

For given row vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n-1}$ and $\boldsymbol{y}_2, \dots, \boldsymbol{y}_{n-1}$, $U(\{\boldsymbol{x}_i\}_{i=1}^{n-1}, \{\boldsymbol{y}_i\}_{i=2}^{n})$ is the matrix on the right hand side of (2.2). The matrix $U'(\cdot)$ is the submatrix of $U(\cdot)$ obtained by means of the deletion of its first column and last row.

The QR factorization of a matrix can be computed by using Givens rotations. Denote by $\mathcal{G}(\gamma)$ the 2×2 complex Givens rotation of parameter $\gamma \in \mathbb{C} \cup \{\infty\}$ given by

$$\begin{aligned} \mathcal{G}(\gamma) &= (\sqrt{1+|\gamma|^2})^{-1} \begin{bmatrix} 1 & \gamma \\ -\bar{\gamma} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \phi & \psi \\ -\bar{\psi} & \phi \end{bmatrix} \gamma, \psi \in \mathbb{C},\ \phi \in \mathbb{R},\ |\psi|^2 + |\phi|^2 = 1, \end{aligned}$$

and

$$\mathcal{G}(\infty) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Observe that it is always possible to fix the value of γ in such a way that $\mathcal{G}(\gamma)$ transforms a vector $[a, b]^T \in \mathbb{C}^2$ into a vector of the form $[\rho, 0]^T$ with $|\rho| = \| [a, b]^T \|_2$. If $a \neq 0$, we set $\bar{\gamma} = b/a$ and otherwise choose $\gamma = \infty$. Then define the $n \times n$ Givens rotation $\mathcal{G}_{k,k+1}(\gamma)$ of parameter γ in coordinates k and $k+1$ by means of

$$\mathcal{G}_{k,k+1}(\gamma) = I_{k-1} \oplus \mathcal{G}(\gamma) \oplus I_{n-k-1} = \begin{bmatrix} I_{k-1} & 0 & 0 \\ 0 & \mathcal{G}(\gamma) & 0 \\ 0 & 0 & I_{n-k-1} \end{bmatrix}.$$

3 Invariance of the structure under the QR iteration

In this section we show that the structure (1.2) of a matrix $A \in \mathbb{C}^{n\times n}$ is maintained under the QR iteration (1.1). We recall that, under quite mild assumptions the sequence $\{A_s\}$ tends to an upper triangular or, at least, a block upper triangular matrix thus yielding information about the eigenvalues of A.

Whenever the matrix R_s in (1.1) is nonsingular, it is easily found that $A_{s+1} = R_s A_s R_s^{-1}$, and this relation allows one to prove that the structure of the lower triangular portion of $A_0 \in \mathcal{C}_n$ is maintained at any step of QR iteration (1.1).

Theorem 3.1. *Let A_s be the matrix obtained in s steps of the QR iteration* (1.1) *applied to $A = A_0 \in \mathcal{C}_n$. For a given integer $\tilde{s} \geq 0$, assume that $R_0, \ldots, R_{\tilde{s}}$ are nonsingular so that, for $0 \leq s \leq \tilde{s}$, we can write $A_{s+1} = R_s A_s R_s^{-1}$. Then each matrix A_s, $0 \leq s \leq \tilde{s}+1$, satisfies*

$$\operatorname{tril}(A_s, -1) = L(\{u_i^{(s)}\}_{i=2}^{n}, \{\overline{v_i^{(s)}}\}_{i=1}^{n-1}, \{t_i^{(s)}\}_{i=2}^{n-1}), \tag{3.1}$$

for suitable numbers $u_2^{(s)}, \ldots, u_n^{(s)}$, $v_1^{(s)}, \ldots, v_{n-1}^{(s)}$ and $t_2^{(s)}, \ldots, t_{n-1}^{(s)}$.

Proof. The proof is by induction on $s \leq \tilde{s}+1$. The case $s = 0$ follows from $A_0 \in \mathcal{C}_n$. Assume that the strictly lower triangular part of A_s, $s \leq \tilde{s}$, is such that (3.1) holds for suitable $u_2^{(s)}, \ldots, u_n^{(s)}$, $v_1^{(s)}, \ldots, v_{n-1}^{(s)}$ and $t_2^{(s)}, \ldots, t_{n-1}^{(s)}$, and then prove the theorem for $s+1$. Write $R_s = (r_{i,j}^{(s)})$, $R_s^{-1} = W_s = (w_{i,j}^{(s)})$ and $A_{s+1/2} = A_s W_s$. $A_{s+1/2}$ is obtained by linearly combining the columns of A_s. Hence, $\operatorname{tril}(A_{s+1/2}, -1)$ admits the following representation:

$$\operatorname{tril}(A_{s+1/2}, -1) = L(\{u_i^{(s+1/2)}\}_{i=2}^{n}, \{\overline{v_i^{(s+1/2)}}\}_{i=1}^{n-1}, \{t_i^{(s+1/2)}\}_{i=2}^{n-1}),$$

where $u_j^{(s+1/2)} = u_j^{(s)}$, $t_j^{(s+1/2)} = t_j^{(s)}$ and, moreover,

$$\begin{aligned} \overline{v_1^{(s+1/2)}} &= w_{1,1}^{(s)} \overline{v_1^{(s)}}, \\ \overline{v_j^{(s+1/2)}} &= \textstyle\sum_{k=2}^{j} w_{k-1,j}^{(s)} t_j^{(s)} \ldots t_k^{(s)} \overline{v_{k-1}^{(s)}} + w_{j,j}^{(s)} \overline{v_j^{(s)}}, \quad j = 2, \ldots, n-1. \end{aligned} \tag{3.2}$$

Analogously, the rows of $A_{s+1} = R_s A_{s+1/2}$ are linear combinations of the rows of $A_{s+1/2}$. In this way, one deduces that $\operatorname{tril}(A_{s+1}, -1)$ can also be represented in a similar form given by

$$\operatorname{tril}(A_{s+1}, -1) = L(\{u_i^{(s+1)}\}_{i=2}^{n}, \{\overline{v_i^{(s+1)}}\}_{i=1}^{n-1}, \{t_i^{(s+1)}\}_{i=2}^{n-1}),$$

where $v_j^{(s+1)} = v_j^{(s+1/2)}$, $t_j^{(s+1)} = t_j^{(s+1/2)} = t_j^{(s)}$ and

$$\begin{aligned} u_n^{(s+1)} &= r_{n,n}^{(s)} u_n^{(s)}, \\ u_{n-j}^{(s+1)} &= \sum_{k=0}^{j-1} r_{n-j,n-k}^{(s)} t_{n-j}^{(s)} \ldots t_{n-k-1}^{(s)} u_{n-k}^{(s)} \\ &\quad + r_{n-j,n-j}^{(s)} u_{n-j}^{(s)}, \quad j = 1, \ldots, n-2. \end{aligned} \tag{3.3}$$

□

Remark 3.2. By the latter theorem, we can represent the strictly lower triangular part of A_{s+1} by replacing $\boldsymbol{u}^{(s)} = [u_1^{(s)}, \ldots, u_n^{(s)}]^T$ and $\boldsymbol{v}^{(s)} = [v_1^{(s)}, \ldots, v_n^{(s)}]$ with $\boldsymbol{u}^{(s+1)} = [u_1^{(s+1)}, \ldots, u_n^{(s+1)}]^T$ and $\boldsymbol{v}^{(s+1)} = [v_1^{(s+1)}, \ldots, v_n^{(s+1)}]^T$ given by (3.3) and (3.2), for $v_j^{(s+1)} = v_j^{(s+1/2)}$, respectively, and having unchanged

the scalars $t_2^{(s+1)} = t_2^{(s)}, \ldots, t_{n-1}^{(s+1)} = t_{n-1}^{(s)}$. Recursively, this defines a representation of $\mathrm{tril}(A_{s+1}, -1)$ of the form

$$\mathrm{tril}(A_{s+1}, -1) = L(\{u_i^{(s+1)}\}_{i=2}^{n}, \{\overline{v_i^{(s+1)}}\}_{i=1}^{n-1}, \{t_i^{(0)}\}_{i=2}^{n-1}),$$

for suitable $u_2^{(s+1)}, \ldots, u_n^{(s+1)}$ and $v_1^{(s+1)}, \ldots, v_{n-1}^{(s+1)}$. If $t_i = t_i^{(0)} \neq 0$ for $i = 2, \ldots, n-1$, then $\mathrm{tril}(A_0, -1)$ is the strictly lower triangular part of a rank-one matrix and, therefore, the same holds for each matrix A_s, i.e.,

$$\begin{aligned}\mathrm{tril}(A_s, -1) &= L(\{u_i^{(s)}\}_{i=2}^{n}, \{\overline{v_i^{(s)}}\}_{i=1}^{n-1}, \{t_i^{(0)}\}_{i=2}^{n-1}) \\ &= L(\{\hat{u}_i^{(s)}\}_{i=2}^{n}, \{\overline{\hat{v}_i^{(s)}}\}_{i=1}^{n-1}, \{1\}_{i=2}^{n-1}).\end{aligned}$$

In other words, the semiseparable representation for the matrix $\mathrm{tril}(A_s, -1)$ is not unique. Different representations are equivalent in the sense that they generate the same matrix but, from a numerical point of view, the sensitivity of the matrix entries with respect to small perturbations of the parameters can vary greatly. Therefore, one may try to obtain a more robust representation of $\mathrm{tril}(A_s, -1)$ by varying the parameters $u_i^{(s)}$, $v_i^{(s)}$ and $t_i^{(s)}$. These issues are also discussed in [55,57,58] and raise the important question of finding a quasiseparable representation of generalized companion matrices which can be updated in an efficient and robust way during the execution of the QR eigenvalue algorithm. The results of the next section confirm that the subclass $\mathcal{C}_n$ provides such an effective representation.

The proof of Theorem 3.1 can be easily generalized to show that the rank structure in the strictly lower triangular part of an invertible quasiseparable matrix $B \in \mathbb{C}^{n\times n}$ is maintained under the QR iteration. In general the same property does not hold for the strictly upper triangular part except for the case where B^H is a small rank perturbation of B or of its inverse that corresponds to an "almost Hermitian" or an "almost unitary" matrix B, respectively. For any matrix $A \in \mathcal{C}_n$ given by (1.2) we find that $B = A - \boldsymbol{z}\boldsymbol{w}^H$ is Hermitian. Thus, $A = A_0$ is a Hermitian matrix plus a rank-one correction. Since for any $s \geq 0$ we have $A_{s+1} = P_s^H A_0 P_s$, where $P_s = Q_0 \ldots Q_s$, it follows that each matrix A_{s+1} generated by the QR iteration applied to A_0 is a Hermitian matrix plus a rank-one perturbation. By combining this observation with Theorem 3.1, we arrive at the following result.

Theorem 3.3. *Let A_s, $s = 1, \ldots, \tilde{s}+1$, be the matrices generated by the QR scheme* (1.1) *starting with $A = A_0 \in \mathcal{C}_n$ of the form* (1.2)*, where $R_0, \ldots, R_{\tilde{s}}$ are assumed to be nonsingular. Then, each A_s, with $0 \leq s \leq \tilde{s}+1$, belongs to $\mathcal{C}_n$. That is, for $0 \leq s \leq \tilde{s}+1$, there exist real numbers $d_1^{(s)}, \ldots, d_n^{(s)}$, complex numbers $t_2^{(s)}, \ldots, t_{n-1}^{(s)}$, and four n-vectors $\boldsymbol{u}^{(s)} = [u_1^{(s)}, \ldots, u_n^{(s)}]^T \in \mathbb{C}^n$, $\boldsymbol{v}^{(s)} = [v_1^{(s)}, \ldots, v_n^{(s)}]^T \in \mathbb{C}^n$, $\boldsymbol{z}^{(s)} = [z_1^{(s)}, \ldots, z_n^{(s)}]^T \in \mathbb{C}^n$ and $\boldsymbol{w}^{(s)} =$*

$[w_1^{(s)}, \ldots, w_n^{(s)}]^T \in \mathbb{C}^n$ *such that* $A_s = (a_{i,j}^{(s)})$ *admits the following representation:*

$$\begin{cases} a_{i,i}^{(s)} = d_i^{(s)} + z_i^{(s)}\overline{w_i^{(s)}}, \quad 1 \le i \le n; \\ a_{i,j}^{(s)} = u_i^{(s)} t_{i,j}^{(s)\times} \overline{v_j^{(s)}}, \quad 1 \le j < i,\ 2 \le i \le n; \\ a_{i,j}^{(s)} = \overline{u_j^{(s)} t_{j,i}^{(s)}}^{\times} v_i^{(s)} + z_i^{(s)}\overline{w_j^{(s)}} - \overline{z_j^{(s)}} w_i^{(s)}, \quad 1 \le i < j,\ 2 \le j \le n, \end{cases} \tag{3.4}$$

where $t_{i,j}^{(s)\times} = t_{i-1}^{(s)} \ldots t_{j+1}^{(s)}$ *for* $i - 1 \ge j + 1$ *and, otherwise,* $t_{i,i-1}^{(s)\times} = 1$.

Proof. For $s = 0$ the claim follows from $A \in \mathcal{C}_n$ by setting $\boldsymbol{u}^{(0)} = \boldsymbol{u}, \boldsymbol{v}^{(0)} = \boldsymbol{v}$, $\boldsymbol{z}^{(0)} = \boldsymbol{z}$, $\boldsymbol{w}^{(0)} = \boldsymbol{w}$, $d_i^{(0)} = d_1$, $1 \le i \le n$, and $t_i^{(0)} = t_i$, $2 \le i \le n - 1$. Recall that $A_0 - \boldsymbol{z}^{(0)}\boldsymbol{w}^{(0)H} = B_0$ is a Hermitian matrix. For $s > 0$ the second equality in (3.4) is established in Theorem 3.1. Moreover, since

$$\begin{aligned} A_{s+1} &= P_s^H A_0 P_s = P_s^H (B_0 + \boldsymbol{z}^{(0)}\boldsymbol{w}^{(0)H}) P_s \\ &= P_s^H B_0 P_s + \boldsymbol{z}^{(s+1)}\boldsymbol{w}^{(s+1)H}, \quad s \ge 0, \end{aligned}$$

where $P_s = Q_0 Q_1 \cdots Q_{s-1}$, we find that A_{s+1} is a rank-one correction of the Hermitian matrix $B_{s+1} = P_s^H B_0 P_s$, for $s = 0, \ldots, \tilde{s}$. From this, we characterize the diagonal and superdiagonal entries of A_s. We first deduce that

$$a_{i,j}^{(s)} - z_i^{(s)}\overline{w_j^{(s)}} = \overline{a_{j,i}^{(s)}} - \overline{z_j^{(s)}} w_i^{(s)}, \quad 1 \le i, j \le n. \tag{3.5}$$

For $i < j$, we have $a_{j,i}^{(s)} = u_j^{(s)} t_{j,i}^{(s)\times} \overline{v_i^{(s)}}$, $\overline{a_{j,i}^{(s)}} = \overline{u_j^{(s)} t_{j,i}^{(s)\times}} v_i^{(s)}$. Substitute the latter expression into (3.5) and obtain

$$a_{i,j}^{(s)} = \overline{u_j^{(s)} t_{j,i}^{(s)}}^{\times} v_i^{(s)} + z_i^{(s)}\overline{w_j^{(s)}} - \overline{z_j^{(s)}} w_i^{(s)}, \quad i \le j.$$

Otherwise, if $i = j$, then from (3.5) one deduces that the imaginary part of $a_{i,i}^{(s)}$ coincides with that of $z_i^{(s)}\overline{w_i^{(s)}}$, and so we can write

$$a_{i,i}^{(s)} = d_i^{(s)} + z_i^{(s)}\overline{w_i^{(s)}}, \quad 1 \le i \le n,$$

for suitable real numbers $d_1^{(s)}, \ldots, d_n^{(s)}$. □

Remark 3.4. In Theorem 3.1 and 3.3 the preservation of the structure for the matrix $A_1 \in \mathcal{C}_n$ is proved under the auxiliary assumption that $A_0 - \sigma_0 I \in \mathcal{C}_n$ is invertible. In the next section the proofs are extended to cover the singular case by means of a constructive approach. Some related issues concerning the invariance of the quasiseparable structure in the singular case are also discussed in [18].

Remark 3.5. From the proof of the latter theorem we find that there exists a representation of A_{s+1} of the form (3.4) such that

$$\boldsymbol{z}^{(s+1)} = P_s^H \boldsymbol{z}^{(0)} = Q_s^H \boldsymbol{z}^{(s)}$$

and

$$\boldsymbol{w}^{(s+1)^H} = \boldsymbol{w}^{(0)^H} P_s = \boldsymbol{w}^{(s)^H} Q_s.$$

These equations provide simple rules for updating the vectors $\boldsymbol{z}^{(s)}$ and $\boldsymbol{w}^{(s)}$ at each step of the QR iteration.

Theorem 3.3 means that the matrices A_s generated at the first iterations in the QR scheme (1.1) applied to $A_0 = A \in \mathcal{C}_n$ inherit the structure (3.4) of their ancestor A_0. If, for a certain index $\hat{s}$, $R_{\hat{s}}$ and $A_{\hat{s}} - \sigma_{\hat{s}} I_n$ are singular, then $\sigma_{\hat{s}}$ is an eigenvalue of A_0, and a deflation technique should be employed. When working in finite precision arithmetic, deflation is also used if the entries of the matrix $A_{\hat{s}}$ satisfy a suitable stopping criterion. Let $A_{\hat{s}}[1\colon n-k, 1\colon n-k] \in \mathbb{C}^{(n-k)\times(n-k)}$ be the leading principal submatrix of $A_{\hat{s}}$ obtained from $A_{\hat{s}}$ by deleting its last k rows and columns. It is easily seen that $A_{\hat{s}}[1\colon n-k, 1\colon n-k]$ admits a representation similar to the one provided by Theorem 3.3. Such a representation is found simply by truncating the corresponding representation of the matrix $A_{\hat{s}}$ of larger size. Hence, $A_{\hat{s}}[1\colon n-k, 1\colon n-k] \in \mathcal{C}_{n-k}$ and, therefore, all the matrices generated by means of the QR scheme (1.1) applied to $A_0 \in \mathcal{C}_n$ for the computation of its eigenvalues still satisfy (3.4).

4 Efficient implementation of the QR iteration

We have already shown that if $A_0 = (a_{i,j}^{(0)}) \in \mathbb{C}^{n\times n}$ is of the form (3.4) then the matrix A_1 generated by the first step of (1.1) admits a similar representation. In this way, the first step of the QR iterative process (1.1) reduces to the computation of real numbers $d_1^{(1)}, \ldots, d_n^{(1)}$, complex numbers $t_2^{(1)}, \ldots, t_{n-1}^{(1)}$, and the entries of the vectors $\boldsymbol{u}^{(1)} \in \mathbb{C}^n$, $\boldsymbol{v}^{(1)} \in \mathbb{C}^n$, $\boldsymbol{z}^{(1)} \in \mathbb{C}^n$ and $\boldsymbol{w}^{(1)} \in \mathbb{C}^n$ which define $A_1 = (a_{i,j}^{(1)})$ according to (3.4). To perform this task efficiently, in this section we investigate the structural properties of the QR factorization of $A_0 - \sigma_0 I_n$ with $A_0 \in \mathcal{C}_n$. For the sake of notational simplicity, without loss of generality, we may assume that $\sigma_0 = 0$.

Recall that $\mathcal{C}_n$ is a subset of the more general class of quasiseparable matrices studied in [23,25,24]. In particular, $A \in \mathcal{C}_n$ is quasiseparable of order $(1, 3)$ with a very special rank structure in its strictly upper triangular part. The unitary factor Q and the upper triangular factor R such that $A = QR$ inherit the quasiseparable structure of A. Numerical methods for computing the generators of the quasiseparable representation of Q and R are provided

in [24], they are based on previous results reported in the book [20]. However, the methods of [24] work for general quasiseparable matrices and thus, for both speed and accuracy reasons, it is important to specialize them for dealing with the special structures considered here. In addition, as we noted in the introduction, while both A_0 and A_1 are quasiseparable of order (1, 3) it turns out that R_0 is quasiseparable of order (0, 4) and, therefore, the explicit computation of the generators of its quasiseparable structure introduces a parametrization which is both mathematically and numerically redundant.

In this section we circumvent these difficulties. We first devise a suitable adaptation of the algorithm in [24] for the QR factorization of quasiseparable matrices A_0 of the form (3.4). The upper triangular matrix R is not explicitly formed but implicitly determined in a partially factored form as the sum of the products of triangular quasiseparable matrices. Then we describe a method for recovering the structural representation (3.4) of the matrix $A_1 = RQ$ from certain quantities generated step-by-step in the process of multiplying R and Q given in their (partially) factored forms.

As usual, the unitary matrix Q_0 can be constructed as a product of Givens rotations suitably chosen to annihilate specific entries of A_0. By exploiting the structure of $\mathrm{tril}(A_0, -1)$, we express Q_0 as the product of $2n-3$ Givens rotations. The following two-step procedure is used to compute a QR factorization $A_0 = Q_0 R_0$ of the matrix $A_0 \in \mathcal{C}_n$. The scheme has been first derived in the paper [24] as a special case of a more general factorization algorithm for finite and infinite quasiseparable matrices suggested in [20].

1) A_0 is reduced to an upper Hessenberg matrix:

$$H_0 = \mathcal{G}_{2,3}(\gamma_{n-2}) \dots \mathcal{G}_{n-1,n}(\gamma_1) A_0; \tag{4.1}$$

2) H_0 is transformed into an upper triangular matrix:

$$R_0 = \mathcal{G}_{n-1,n}(\gamma_{2n-3}) \dots \mathcal{G}_{1,2}(\gamma_{n-1}) H_0. \tag{4.2}$$

From (4.1) and (4.2) we find that

$$R_0 = \mathcal{G}_{n-1,n}(\gamma_{2n-3}) \dots \mathcal{G}_{1,2}(\gamma_{n-1}) \mathcal{G}_{2,3}(\gamma_{n-2}) \dots \mathcal{G}_{n-1,n}(\gamma_1) A_0.$$

It follows that

$$Q_0^H = \mathcal{G}_{n-1,n}(\gamma_{2n-3}) \dots \mathcal{G}_{1,2}(\gamma_{n-1}) \mathcal{G}_{2,3}(\gamma_{n-2}) \dots \mathcal{G}_{n-1,n}(\gamma_1) \tag{4.3}$$

is the desired unitary matrix such that $Q_0^H A_0 = R_0$.

Once the upper triangular factor R_0 and the Givens rotations $\mathcal{G}_{j,j+1}(\gamma_k)$ are known, the matrix A_1 can be determined by

$$A_1 = R_0 (\mathcal{G}_{n-1,n}(\gamma_1))^H \dots (\mathcal{G}_{2,3}(\gamma_{n-2}))^H (\mathcal{G}_{1,2}(\gamma_{n-1}))^H \dots (\mathcal{G}_{n-1,n}(\gamma_{2n-3}))^H.$$

The composite scheme for the transition from A_0 to A_1 can be summarized as follows:

$$A_0 \stackrel{1}{\to} H_0 \stackrel{2}{\to} R_0 \stackrel{3}{\to} A_1. \tag{4.4}$$

We prove that both H_0 and R_0 can be described in terms of $O(n)$ parameters; moreover, the matrix structures can be managed by means of efficient and numerically stable methods. In addition, Q_0 does not need to be formed explicitly but it can be implicitly defined as the product of $2n-3$ Givens rotation matrices $\mathcal{G}_{j,j+1}(\gamma_k)$. Combining these facts enables us to devise a robust algorithm for the scheme (4.4) having linear complexity.

The remaining part of this section is divided into three subsections, each of them is devoted to a single step in (4.4).

4.1 Reduction to the Hessenberg form

The upper Hessenberg matrix H_0 is defined by (4.1), where the Givens rotation matrices $\mathcal{G}_{2,3}(\gamma_{n-2}), \ldots, \mathcal{G}_{n-1,n}(\gamma_1)$ are chosen to zero the respective entries of $\mathrm{tril}(A_0, -2)$. To do this, the parameters γ_j can be determined as follows. Choose the first element γ_1 to yield

$$\mathcal{G}(\gamma_1)\begin{bmatrix} u_{n-1}^{(0)} \\ u_n^{(0)} t_{n-1}^{(0)} \end{bmatrix} = \begin{bmatrix} \widehat{u}_{n-1}^{(0)} \\ 0 \end{bmatrix}. \tag{4.5}$$

Similarly, choose the successive entries $\gamma_2, \ldots, \gamma_{n-2}$ to yield

$$\mathcal{G}(\gamma_j)\begin{bmatrix} u_{n-j}^{(0)} \\ \widehat{u}_{n-j+1}^{(0)} t_{n-j}^{(0)} \end{bmatrix} = \begin{bmatrix} \widehat{u}_{n-j}^{(0)} \\ 0 \end{bmatrix}, \quad 2 \le j \le n-2. \tag{4.6}$$

To describe the effects of pre-multiplying A_0 by $\mathcal{G}_{n-1,n}(\gamma_1), \ldots, \mathcal{G}_{2,3}(\gamma_{n-2})$ we first investigate the structure of the unitary matrix

$$\widehat{Q}_0 = \mathcal{G}_{2,3}(\gamma_{n-2}) \ldots \mathcal{G}_{n-1,n}(\gamma_1) = \left[\begin{array}{c|c} 1 & \mathbf{0}^T \\ \hline \mathbf{0} & \widehat{Q} \end{array}\right]. \tag{4.7}$$

It is easy to verify that $\widehat{Q}_0$ is an upper Hessenberg matrix. The following result proved in [36] also reveals the quasiseparable structure of its trailing principal submatrix $\widehat{Q}$.

Theorem 4.1. *The matrix $\widehat{Q} \in \mathbb{C}^{(n-1)\times(n-1)}$ in (4.7) admits the following representation:*

$$\widehat{Q} = \begin{bmatrix} \tilde{\phi}\phi_{n-2} & \tilde{\phi}\psi_{n-2}\phi_{n-3} & \ldots & \ldots & \tilde{\phi}\psi_{n-2}\ldots\psi_1\hat{\phi} \\ -\overline{\psi}_{n-2} & \phi_{n-2}\phi_{n-3} & \ldots & \ldots & \phi_{n-2}\psi_{n-3}\ldots\psi_1\hat{\phi} \\ 0 & -\overline{\psi}_{n-3} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & -\overline{\psi}_1 & \phi_1\hat{\phi} \end{bmatrix}, \tag{4.8}$$

where for the sake of notational simplicity we may write $\tilde{\phi} = \hat{\phi} = 1$.

Write $\boldsymbol{x}_i^{(0)} = [z_i^{(0)}, w_i^{(0)}]$ and $\boldsymbol{y}_i^{(0)} = [w_i^{(0)}, -z_i^{(0)}]$ for $i = 1, \ldots, n$, and

$$A_0 = B_0 + C_0,$$

where $B_0 = (b_{i,j}^{(0)}) \in \mathbb{C}^{n\times n}$ and $C_0 = (c_{i,j}^{(0)}) \in \mathbb{C}^{n\times n}$ are defined by

$$\begin{aligned} B_0 &= L(\{u_i^{(0)}\}_{i=2}^n, \{\overline{v_i^{(0)}}\}_{i=1}^{n-1}, \{t_i^{(0)}\}_{i=2}^{n-1}) + \mathrm{diag}(\{a_{i,i}^{(0)}\}_{i=1}^n) \\ &\quad + U(\{\boldsymbol{x}_i^{(0)}\}_{i=1}^{n-1}, \{\boldsymbol{y}_i^{(0)}\}_{i=2}^n), \end{aligned} \tag{4.9}$$

and

$$C_0 = R(\{\overline{u_i^{(0)}}\}_{i=2}^n, \{v_i^{(0)}\}_{i=1}^{n-1}, \{\overline{t_i^{(0)}}\}_{i=2}^{n-1}), \tag{4.10}$$

respectively. Now observe that

$$\widehat{Q}_0 A_0 = \widehat{Q}_0 B_0 + \widehat{Q}_0 C_0,$$

The following result specifies the structure of $\widehat{Q}_0 B_0$. The proof follows from [[24], Theorem 6.1, p. 430].

Theorem 4.2. *The matrix* $\widehat{Q}_0 B_0$ *is defined as follows:*

$$\widehat{Q}_0 B_0 = \left[\begin{array}{c|ccc} a_{1,1}^{(0)} & \boldsymbol{x}_1^{(0)}\boldsymbol{y}_2^{(0)H} & \cdots & \boldsymbol{x}_1^{(0)}\boldsymbol{y}_n^{(0)H} \\ \hline \widehat{u}_2^{(0)}\overline{v_1^{(0)}} & & & \\ 0 & & \widehat{B}_0 & \\ \vdots & & & \end{array}\right],$$

where

$$\begin{aligned} \widehat{B}_0 &= \mathrm{Subdiag}(\eta_2, \ldots, \eta_{n-1}) + U'(\{\widehat{\boldsymbol{x}}_i^{(0)}\}_{i=2}^n, \{\boldsymbol{y}_i^{(0)}\}_{i=2}^n) \\ &\quad + R'(\{\zeta_i\}_{i=2}^n, \{\phi_{n-i}\}_{i=1}^{n-1}, \{\psi_{n-i}\}_{i=2}^{n-1}). \end{aligned}$$

That is, we have

$$\begin{aligned} \widehat{B}_0 &= \begin{bmatrix} \widehat{\boldsymbol{x}}_2^{(0)}\boldsymbol{y}_2^{(0)H} & \cdots & \cdots & \widehat{\boldsymbol{x}}_2^{(0)}\boldsymbol{y}_n^{(0)H} \\ \eta_2 & \widehat{\boldsymbol{x}}_3^{(0)}\boldsymbol{y}_3^{(0)H} & \cdots & \widehat{\boldsymbol{x}}_3^{(0)}\boldsymbol{y}_n^{(0)H} \\ & \ddots & \ddots & \vdots \\ 0 & \cdots & \eta_{n-1} & \widehat{\boldsymbol{x}}_n^{(0)}\boldsymbol{y}_n^{(0)H} \end{bmatrix} \\ &\quad + \begin{bmatrix} \phi_{n-1}\zeta_2 & \phi_{n-1}\psi_{n-2}\zeta_3 & \cdots & \phi_{n-1}\psi_{n-2}\cdots\psi_1\zeta_n \\ 0 & \zeta_3\phi_{n-2} & \cdots & \phi_{n-2}\psi_{n-3}\cdots\psi_1\zeta_n \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \phi_1\zeta_n \end{bmatrix}, \quad (\phi_{n-1} = 1). \end{aligned} \tag{4.11}$$

The subdiagonal entries η_j are defined by

$$\eta_j = -\overline{\psi_{n-j}} a^{(0)}_{j,j} + \phi_{n-j} \widehat{u}^{(0)}_{j+1} \overline{v^{(0)}_j}, \quad j = 2, \ldots, n-1. \tag{4.12}$$

The remaining elements ζ_j are determined according to the following equation:

$$\begin{aligned} &\zeta_j = \phi_{n-j} a^{(0)}_{j,j} + \psi_{n-j} \widehat{u}^{(0)}_{j+1} \overline{v^{(0)}_j} - \tilde{\boldsymbol{x}}^{(0)}_j \boldsymbol{y}^{(0)H}_j, \quad j = 2, \ldots, n-1, \\ &\zeta_n = a^{(0)}_{n,n}, \end{aligned} \tag{4.13}$$

where $\tilde{\boldsymbol{x}}^{(0)}_j$ and $\widehat{\boldsymbol{x}}^{(0)}_j$ are generated by the following two-step procedure subjected to the initialization $\tilde{\boldsymbol{x}}^{(0)}_n = \mathbf{0}$:

for $j = 1 \,:\, n-2$

1. $\widehat{\boldsymbol{x}}^{(0)}_{n-j+1} = -\overline{\psi_j} \boldsymbol{x}^{(0)}_{n-j} + \phi_j \tilde{\boldsymbol{x}}^{(0)}_{n-j+1}$;
2. $\tilde{\boldsymbol{x}}^{(0)}_{n-j} = \phi_j \boldsymbol{x}^{(0)}_{n-j} + \psi_j \tilde{\boldsymbol{x}}^{(0)}_{n-j+1}$;

end

$\widehat{\boldsymbol{x}}^{(0)}_2 = \tilde{\boldsymbol{x}}^{(0)}_2$.

Summing up, we represent the matrix $\widehat{Q}_0 B_0$ by means of a data structure of linear size whose elements are computed at a linear cost. The same clearly holds for the upper Hessenberg matrix H_0 conveniently described as

$$H_0 = \left[\begin{array}{c|ccc} a^{(0)}_{1,1} & \boldsymbol{x}^{(0)}_1 \boldsymbol{y}^{(0)H}_2 & \cdots & \boldsymbol{x}^{(0)}_1 \boldsymbol{y}^{(0)H}_n \\ \hline \widehat{u}^{(0)}_2 \overline{v^{(0)}_1} & & & \\ 0 & & \widehat{B}_0 & \\ \vdots & & & \end{array}\right] + \left[\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \widehat{Q} \end{array}\right] C_0, \tag{4.14}$$

where $\widehat{B}_0$, $\widehat{Q}$ and C_0 are given by (4.11), (4.8) and (4.10), respectively.

4.2 Reduction to the upper triangular form

We reduce the matrix $H_0 = (h^{(0)}_{i,j})$ of (4.14) to the upper triangular form $R_0 = (r^{(0)}_{i,j})$ by applying Givens rotations $\mathcal{G}_{n-1,n}(\gamma_{2n-3}), \ldots, \mathcal{G}_{1,2}(\gamma_{n-1})$ to annihilate the entries located on the first subdiagonal. The first Givens rotation $\mathcal{G}_{1,2}(\gamma_{n-1})$ is determined by the vector equation

$$\mathcal{G}(\gamma_{n-1}) \left[\begin{array}{c} a^{(0)}_{1,1} \\ \hline \widehat{u}^{(0)}_2 \overline{v^{(0)}_1} \end{array}\right] = \left[\begin{array}{c} r^{(0)}_{1,1} \\ 0 \end{array}\right].$$

If we pre-multiply H_0 by $\mathcal{G}_{1,2}(\gamma_{n-1})$, we obtain that

$$\mathcal{G}_{1,2}(\gamma_{n-1})H_0 = \left[\begin{array}{c|c} r_{1,1}^{(0)} & \widetilde{\boldsymbol{x}}_1^{(0)}\boldsymbol{y}_2^{(0)} + \psi_{n-1}\zeta_2 \ \ldots \ \widetilde{\boldsymbol{x}}_1^{(0)}\boldsymbol{y}_n^{(0)H} + \psi_{n-1}..\psi_1\zeta_n \\ \hline \mathbf{0} & \widetilde{B}_0 \end{array}\right] + \widetilde{Q}_0 C_0,$$

where $\widetilde{\boldsymbol{x}}_1^{(0)} = \phi_{n-1}\boldsymbol{x}_1^{(0)} + \psi_{n-1}\widehat{\boldsymbol{x}}_2^{(0)}$,

$$\widetilde{B}_0 = \left[\begin{array}{cccc} \widehat{\boldsymbol{x}}_2^{(0)}\boldsymbol{y}_2^{(0)H} & \ldots & \ldots & \widehat{\boldsymbol{x}}_2^{(0)}\boldsymbol{y}_n^{(0)H} \\ \eta_2 & \widehat{\boldsymbol{x}}_3^{(0)}\boldsymbol{y}_3^{(0)H} & \ldots & \widehat{\boldsymbol{x}}_3^{(0)}\boldsymbol{y}_n^{(0)H} \\ & \ddots & \ddots & \vdots \\ 0 & \ldots & \eta_{n-1} & \widehat{\boldsymbol{x}}_n^{(0)}\boldsymbol{y}_n^{(0)H} \end{array}\right] + \left[\begin{array}{cccc} \phi_{n-1}\zeta_2 & \phi_{n-1}\psi_{n-2}\zeta_3 & \ldots & \phi_{n-1}\psi_{n-2}\ldots\psi_1\zeta_n \\ 0 & \zeta_3\phi_{n-2} & \ldots & \phi_{n-2}\psi_{n-3}\ldots\psi_1\zeta_n \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \phi_1\zeta_n \end{array}\right],$$

with $\widehat{\boldsymbol{x}}_2^{(0)} = \phi_{n-1}\widehat{\boldsymbol{x}}_2^{(0)} - \overline{\psi_{n-1}}\boldsymbol{x}_1^{(0)}$, and, moreover,

$$\widetilde{Q}_0 = \left[\begin{array}{ccccc} \tilde{\phi}\phi_{n-1} & \tilde{\phi}\psi_{n-1}\phi_{n-2} & \ldots & \ldots & \tilde{\phi}\psi_{n-1}\ldots\psi_1\hat{\phi} \\ -\overline{\psi_{n-1}} & \phi_{n-1}\phi_{n-2} & \ldots & \ldots & \phi_{n-1}\psi_{n-2}\ldots\psi_1\hat{\phi} \\ 0 & -\overline{\psi}_{n-2} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & -\overline{\psi}_1 & \phi_1\hat{\phi} \end{array}\right],$$

with $\tilde{\phi} = \hat{\phi} = 1$.

We have

$$\widetilde{Q}_0 = \mathrm{Subdiag}(\{-\overline{\psi_{n-i}}\}_{i=1}^{n-1}) + R'(\{\phi_{n-i}\}_{i=1}^{n-1} \odot \{1\}, \{1\} \odot \{\phi_{n-i}\}_{i=1}^{n-1}, \{\psi_{n-i}\}_{i=1}^{n-1}),$$

and

$$\widetilde{B}_0 = \mathrm{Subdiag}(\{\eta_j\}_{j=2}^{n-1}) + R'(\{\zeta_i\}_{i=2}^{n}, \{\phi_{n-i}\}_{i=1}^{n-1}, \{\psi_{n-i}\}_{i=1}^{n-1}) + U'(\{\widehat{\boldsymbol{x}}_i^{(0)}\}_{i=2}^{n}, \{\boldsymbol{y}_i^{(0)}\}_{i=2}^{n}).$$

Once $\widetilde{B}_0$ and $\widetilde{Q}_0$ have been computed, we may determine γ_n and then continue the triangularization process. The following theorem is a suitable specialization of Theorem 6.3 in [24] and provides the desired representation of the upper triangular matrix R_0 obtained at the end of the triangularization process applied to H_0.

Theorem 4.3. *The matrix* $R_0 = \mathcal{G}_{n-1,n}(\gamma_{2n-3}) \dots \mathcal{G}_{1,2}(\gamma_{n-1})\mathcal{G}_{2,3}(\gamma_{n-2}) \dots \mathcal{G}_{n-1,n}(\gamma_1)A_0 = \mathcal{G}_{n-1,n}(\gamma_{2n-3}) \dots \mathcal{G}_{1,2}(\gamma_{n-1})H_0$ *admits the following representation:*

$$R_0 = R_0^{(1)} + R_0^{(2)} + Q_0^{(1)}C_0 + Q_0^{(2)}C_0,$$

where

$$R_0^{(1)} = \begin{bmatrix} r_{1,1}^{(0)}\,\widetilde{\boldsymbol{x}}_1^{(0)}\boldsymbol{y}_2^{(0)H} + \delta_1\zeta_2 & \dots & \widetilde{\boldsymbol{x}}_1^{(0)}\boldsymbol{y}_n^{(0)H} + \delta_1\psi_{n-2}\dots\psi_1\zeta_n \\ 0 & r_{2,2}^{(0)} - \phi_n\widetilde{s}_2 & \dots & \widetilde{\boldsymbol{x}}_2^{(0)}\boldsymbol{y}_n^{(0)H} + \delta_2\psi_{n-3}\dots\psi_1\zeta_n \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \widetilde{\boldsymbol{x}}_n^{(0)}\boldsymbol{y}_n^{(0)H} + \delta_n\zeta_n \end{bmatrix},$$

$$R_0^{(2)} = \begin{bmatrix} 0 & & & & & \\ 0 & 0 & & & & \\ \vdots & \overline{\psi_n}\phi_{n+1}\widetilde{s}_2 & \ddots & & & \\ \vdots & -\widetilde{s}_2\phi_{n+2}\prod_{j=n}^{n+1}(-\overline{\psi_j})\ \overline{\psi_{n+1}}\phi_{n+2}\widetilde{s}_3 & \ddots & & \\ \vdots & \vdots & & \ddots & \ddots & \\ 0 & -\widetilde{s}_2\tilde{\phi}\prod_{j=n}^{2n-3}(-\overline{\psi_j}) & \dots & \dots\ \overline{\psi_{2n-3}}\tilde{\phi}\widetilde{s}_{n-1} & 0 \end{bmatrix},$$

$$Q_0^{(1)} = \begin{bmatrix} \rho_1 & \delta_1\phi_{n-2} & \dots & \delta_1\psi_{n-2}\dots\psi_1\tilde{\phi} \\ & \ddots & \ddots & \vdots \\ & & \rho_{n-1} & \delta_{n-1}\tilde{\phi} \\ 0 & \dots & 0 & \rho_n \end{bmatrix},$$

and

$$Q_0^{(2)} = \begin{bmatrix} & 0 & & \\ & \phi_n\mu_1 & 0 & \\ & \vdots & \ddots & \ddots \\ & & & \\ \phi_{2n-3}(-\overline{\psi_n})\dots(-\overline{\psi_{2n-4}})\mu_1 & \dots & \phi_{2n-3}\mu_{n-2} & 0 \\ \tilde{\phi}(-\overline{\psi_n})\dots(-\overline{\psi_{2n-3}})\mu_1 & \dots & \dots & \tilde{\phi}\mu_{n-1}\ 0 \end{bmatrix}, \quad (\tilde{\phi} = 1).$$

The vectors $\widetilde{\boldsymbol{x}}_j^{(0)}$ *and* $\check{\boldsymbol{x}}_j^{(0)}$ *are generated by the following two-step procedure subjected to the initializations* $\widetilde{\boldsymbol{x}}_1^{(0)} = \phi_{n-1}\boldsymbol{x}_1^{(0)} + \psi_{n-1}\widehat{\boldsymbol{x}}_2^{(0)}$ *and* $\check{\boldsymbol{x}}_2^{(0)} = \widetilde{\boldsymbol{x}}_2^{(0)} = \phi_{n-1}\widehat{\boldsymbol{x}}_2^{(0)} - \overline{\psi_{n-1}}\boldsymbol{x}_1^{(0)}$*:*

for $j = 2 : n-1$

1. $\widetilde{\boldsymbol{x}}_j^{(0)} = \phi_{n-2+j}\check{\boldsymbol{x}}_j^{(0)} + \psi_{n-2+j}\widehat{\boldsymbol{x}}_{j+1}^{(0)}$;
2. $\check{\boldsymbol{x}}_{j+1}^{(0)} = \phi_{n-2+j}\widehat{\boldsymbol{x}}_{j+1}^{(0)} - \overline{\psi_{n-2+j}}\check{\boldsymbol{x}}_j^{(0)}$;

end

$\widetilde{\boldsymbol{x}}_n^{(0)} = \check{\boldsymbol{x}}_n^{(0)}$.

The parameters ϕ_j and ψ_j, $n-1 \le j \le 2n-3$, defining the Givens rotations $\mathcal{G}_{n-1,n}(\gamma_{2n-3}), \ldots, \mathcal{G}_{1,2}(\gamma_{n-1})$, are determined by the following vector equations:

$$\mathcal{G}(\gamma_{n-1}) \begin{bmatrix} a_{1,1}^{(0)} \\ \widehat{u}_2^{(0)} \overline{v_1^{(0)}} \end{bmatrix} = \begin{bmatrix} r_{1,1}^{(0)} \\ 0 \end{bmatrix}$$

and

$$\mathcal{G}(\gamma_{n-2+j}) \begin{bmatrix} \check{\boldsymbol{x}}_j^{(0)} \boldsymbol{y}_j^{(0)^H} + \widehat{\delta}_j \zeta_j + \widetilde{s}_j \\ \eta_j \end{bmatrix} = \begin{bmatrix} r_{j,j}^{(0)} \\ 0 \end{bmatrix}, \quad j = 2, \ldots, n-1,$$

where the quantities δ_j, μ_j, ρ_j, s_j and $\widetilde{s}_j$ are defined according to the following procedures.

The entries δ_j are generated by the following two-step procedure subjected to the initialization $\delta_1 = \psi_{n-1}$, $\widehat{\delta}_2 = \boldsymbol{\phi}_{n-1}$:

for $j = 2 : n-1$

1. $\delta_j = \phi_{n-2+j} \widehat{\delta}_j \psi_{n-j} + \psi_{n-2+j} \phi_{n-j}$;
2. $\widehat{\delta}_{j+1} = -\overline{\psi_{n-2+j}}\, \widehat{\delta}_j \psi_{n-j} + \phi_{n-2+j} \phi_{n-j}$;

end

$\delta_n = \widehat{\delta}_n$.

The parameters ρ_j and μ_j satisfy $\rho_1 = \phi_{n-1}$, $\rho_j = \phi_{n-2+j} \widehat{\delta}_j \phi_{n-j} - \overline{\psi_{n-2+j}}\, \overline{\psi_{n-j}}$, $2 \le j \le n-1$, $\rho_n = \widehat{\delta}_n$, $\mu_1 = -\overline{\psi_{n-1}}$ and $\mu_j = -\overline{\psi_{n-j}} \phi_{n-2+j} - \overline{\psi_{n-2+j}}\, \widehat{\delta}_j \phi_{n-j}$, $2 \le j \le n-1$.

Finally, the elements s_j and $\widetilde{s}_j$ are given by $\widetilde{s}_j = \overline{u_j^{(0)}} s_j$, $2 \le j \le n-1$, where

$$\begin{cases} s_2 = \mu_1 v_1^{(0)}, \\ s_j = (-\overline{\psi_{n-3+j}}) \overline{t_{j-1}^{(0)}} s_{j-1} + \mu_{j-1} v_{j-1}^{(0)}, \quad j = 3, \ldots, n-1. \end{cases}$$

Theorem 4.3 provides a structural description of the matrix R_0 obtained by triangularization of the matrix $A_0 \in \mathcal{C}_n$. The next step of the QR scheme (1.1) is to compute the matrix $A_1 = R_0 Q_0$. From the results of the previous section we know that $A_1 \in \mathcal{C}_n$. Hence, our task reduces to the computation of $d_1^{(1)}, \ldots, d_n^{(1)}, t_2^{(1)}, \ldots, t_{n-1}^{(1)}, \boldsymbol{u}^{(1)} \in \mathbb{C}^n$, $\boldsymbol{v}^{(1)} \in \mathbb{C}^n$, $\boldsymbol{z}^{(1)} \in \mathbb{C}^n$ and $\boldsymbol{w}^{(1)} \in \mathbb{C}^n$, which define $A_1 = (a_{i,j}^{(1)})$ according to (3.4). From Remark 3.5 it follows that

$$\boldsymbol{z}^{(1)} = \mathcal{G}_{n-1,n}(\gamma_{2n-3}) \ldots \mathcal{G}_{1,2}(\gamma_{n-1}) \mathcal{G}_{2,3}(\gamma_{n-2}) \ldots \mathcal{G}_{n-1,n}(\gamma_1) \boldsymbol{z}^{(0)},$$

and, analogously,

$$\boldsymbol{w}^{(1)} = \mathcal{G}_{n-1,n}(\gamma_{2n-3}) \ldots \mathcal{G}_{1,2}(\gamma_{n-1}) \mathcal{G}_{2,3}(\gamma_{n-2}) \ldots \mathcal{G}_{n-1,n}(\gamma_1) \boldsymbol{w}^{(0)}.$$

In this way, it remains to find only the quantities $d_1^{(1)}, \ldots, d_n^{(1)}, t_2^{(1)}, \ldots, t_{n-1}^{(1)}$, $\boldsymbol{u}^{(1)} \in \mathbb{C}^n$ and $\boldsymbol{v}^{(1)} \in \mathbb{C}^n$, which specify the lower triangular part of A_1. This issue is addressed in the next subsection.

4.3 Computation of the new iterate

We have already shown how to calculate the upper triangular matrix R_0 and the unitary matrix Q_0 such that $A_0 = Q_0 R_0$. Once this factorization is known, the QR iteration (1.1) determines the new iterate A_1 as $A_1 = R_0 Q_0$. Observe that

$$\begin{aligned} A_1 &= R_0(\mathcal{G}_{n-1,n}(\gamma_1))^H \ldots (\mathcal{G}_{2,3}(\gamma_{n-2}))^H (\mathcal{G}_{1,2}(\gamma_{n-1}))^H \ldots (\mathcal{G}_{n-1,n}(\gamma_{2n-3}))^H \\ &= A_{1/2}(\mathcal{G}_{1,2}(\gamma_{n-1}))^H \ldots (\mathcal{G}_{n-1,n}(\gamma_{2n-3}))^H, \end{aligned}$$

with

$$A_{1/2} = (a_{i,j}^{1/2}) = R_0(\mathcal{G}_{n-1,n}(\gamma_1))^H \ldots (\mathcal{G}_{2,3}(\gamma_{n-2}))^H. \tag{4.15}$$

Let $\widehat{r}_{j,j}^{(0)}$ denote the diagonal entry of $R_0(\mathcal{G}_{n-1,n}(\gamma_1))^H \ldots (\mathcal{G}_{j,j+1}(\gamma_{n-j}))^H$ in position (j, j), where $\widehat{r}_{n,n}^{(0)} = r_{n,n}^{(0)}$ and $\widehat{r}_{1,1}^{(0)} = r_{1,1}^{(0)}$. The process of forming the matrix $A_{1/2}$ is demonstrated below for a generic 5×5 matrix.

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \widehat{r}_{5,5}^{(0)} \end{bmatrix} \overset{\mathcal{G}_{4,5}(\gamma_1)^H}{\rightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \widehat{r}_{4,4}^{(0)} & \times \\ 0 & 0 & 0 & \overline{\psi}_1 \widehat{r}_{5,5}^{(0)} & \phi_1 \widehat{r}_{5,5}^{(0)} \end{bmatrix} \overset{\mathcal{G}_{3,4}(\gamma_2)^H}{\rightarrow}$$

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \widehat{r}_{3,3}^{(0)} & \times & \times \\ 0 & 0 & \overline{\psi}_2 \widehat{r}_{4,4}^{(0)} & \phi_2 \widehat{r}_{4,4}^{(0)} & \times \\ 0 & 0 & \overline{\psi_1 \psi_2} \widehat{r}_{5,5}^{(0)} & \overline{\psi}_1 \phi_2 \widehat{r}_{5,5}^{(0)} & \phi_1 \widehat{r}_{5,5}^{(0)} \end{bmatrix} \overset{\mathcal{G}_{2,3}(\gamma_3)^H}{\rightarrow} \tag{4.16}$$

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \widehat{r}_{2,2}^{(0)} & \times & \times & \times \\ 0 & \overline{\psi}_3 \widehat{r}_{3,3}^{(0)} & \phi_3 \widehat{r}_{3,3}^{(0)} & \times & \times \\ 0 & \overline{\psi_2 \psi_3} \widehat{r}_{4,4}^{(0)} & \overline{\psi}_2 \phi_3 \widehat{r}_{4,4}^{(0)} & \phi_2 \widehat{r}_{4,4}^{(0)} & \times \\ 0 & \overline{\psi_1 \psi_2 \psi_3} \widehat{r}_{5,5}^{(0)} & \overline{\psi_1 \psi_2} \phi_3 \widehat{r}_{5,5}^{(0)} & \overline{\psi}_1 \phi_2 \widehat{r}_{5,5}^{(0)} & \phi_1 \widehat{r}_{5,5}^{(0)} \end{bmatrix}.$$

This is extended to the next theorem.

Theorem 4.4. *The matrix* $A_{1/2}$ *of* (4.15) *satisfies*

$$\begin{aligned}
&\mathrm{tril}(A_{1/2},1)\\
&= \left[\begin{array}{c|c} \widehat{r}^{(0)}_{1,1} & a^{(1/2)}_{1,2}\ 0\ \dots\ \dots\ \dots\ 0 \\ \hline \mathbf{0} & \left|\mathrm{Subdiag}(\{a^{(1/2)}_{i,i+1}\}_{i=2}^{n-1})^T + L'(\{\widehat{r}^{(0)}_{i,i}\}_{i=2}^{n},\{1\}\odot\{\phi_{n-i}\}_{i=2}^{n-1},\{\overline{\psi}_{n-i}\}_{i=2}^{n-1})\right. \end{array}\right]\\
&= \begin{bmatrix} \widehat{r}^{(0)}_{1,1} & a^{(1/2)}_{1,2} & & & \\ 0 & \widehat{r}^{(0)}_{2,2} & a^{(1/2)}_{2,3} & & \\ 0 & \overline{\psi_{n-2}}\widehat{r}^{(0)}_{3,3} & \phi_{n-2}\widehat{r}^{(0)}_{3,3} & \ddots & \\ \vdots & \vdots & \ddots & \ddots & a^{(1/2)}_{n-1,n} \\ 0 & \overline{\psi_{n-2}\dots\psi_1}\widehat{r}^{(0)}_{n,n} & \dots & \overline{\psi_1}\phi_2\widehat{r}^{(0)}_{n,n} & \phi_1\widehat{r}^{(0)}_{n,n} \end{bmatrix}.
\end{aligned}$$

The transformation from $A_{1/2}$ to A_1 modifies the lower triangular part of $A_{1/2}$ and is next demonstrated in the case of a 5×5 matrix $A_{1/2}$ represented as at the end of the process (4.16). The first two steps are:

$$\begin{aligned}
A_{1/2} &\overset{\mathcal{G}_{1,2}(\gamma_4)^H}{\rightarrow} \begin{bmatrix} a^{(1)}_{1,1} & \times & \times & \times & \times \\ \overline{\psi}_4\widehat{r}^{(0)}_{2,2} & \phi_4\widehat{r}^{(0)}_{2,2} & a^{(1/2)}_{2,3} & \times & \times \\ \overline{\psi_3\psi_4}\widehat{r}^{(0)}_{3,3} & \overline{\psi}_3\phi_4\widehat{r}^{(0)}_{3,3} & \phi_3\widehat{r}^{(0)}_{3,3} & \times & \times \\ \overline{\psi_2\psi_3\psi_4}\widehat{r}^{(0)}_{4,4} & \overline{\psi_2\psi_3}\phi_4\widehat{r}^{(0)}_{4,4} & \overline{\psi_2}\phi_3\widehat{r}^{(0)}_{4,4} & \times & \times \\ \overline{\psi_1\psi_2\psi_3\psi_4}\widehat{r}^{(0)}_{5,5} & \overline{\psi_1\psi_2\psi_3}\phi_4\widehat{r}^{(0)}_{5,5} & \overline{\psi_1\psi_2}\phi_3\widehat{r}^{(0)}_{4,4} & \times & \times \end{bmatrix}\\
&\overset{\mathcal{G}_{2,3}(\gamma_5)^H}{\rightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & a^{(1)}_{2,2} & \times & \times & \times \\ \times & \overline{v^{(1)}_2}\widehat{r}^{(0)}_{3,3} & \eta_3\widehat{r}^{(0)}_{3,3} & a^{(1/2)}_{3,4} & \times \\ \times & \overline{v^{(1)}_2}\overline{\psi_2}\widehat{r}^{(0)}_{4,4} & \overline{\psi_2}\eta_3\widehat{r}^{(0)}_{4,4} & \phi_2\widehat{r}^{(0)}_{4,4} & \times \\ \times & \overline{v^{(1)}_2}\overline{\psi_1\psi_2}\widehat{r}^{(0)}_{4,4} & \overline{\psi_1\psi_2}\eta_3\widehat{r}^{(0)}_{5,5} & \overline{\psi_1}\phi_2\widehat{r}^{(0)}_{5,5} & \times \end{bmatrix},
\end{aligned}$$

where

$$a^{(1)}_{1,1} = \phi_4\widehat{r}^{(0)}_{1,1} + \overline{\psi}_4 a^{(1/2)}_{1,2}, \quad a^{(1)}_{2,2} = \phi_5\phi_4\widehat{r}^{(0)}_{2,2} + \overline{\psi}_5 a^{(1/2)}_{2,3},$$

and

$$v^{(1)}_1 = \psi_4, \quad , v^{(1)}_2 = \phi_4\phi_5\psi_3 + \psi_5\phi_3, \quad \eta_3 = -\psi_5\overline{\psi_3}\phi_4 + \phi_5\phi_3.$$

The process is completed as follows:

$$A_{1/2}\mathcal{G}_{1,2}(\gamma_4)^H\mathcal{G}_{2,3}(\gamma_5)^H \overset{\mathcal{G}_{3,4}(\gamma_6)^H}{\to} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & a_{3,3}^{(1)} & \times & \times \\ \times & \times & \overline{v_3^{(1)}}\widehat{r}_{4,4}^{(0)} & \eta_4\widehat{r}_{4,4}^{(0)} & a_{4,5}^{(1/2)} \\ \times & \times & \overline{v_3^{(1)}}\overline{\psi_1}\widehat{r}_{5,5}^{(0)} & \eta_4\overline{\psi_1}\widehat{r}_{5,5}^{(0)} & \phi_1\widehat{r}_{5,5}^{(0)} \end{bmatrix}$$

$$\overset{\mathcal{G}_{4,5}(\gamma_7)^H}{\to} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & a_{4,4}^{(1)} & \times \\ \times & \times & \times & \overline{v_4^{(1)}}\widehat{r}_{5,5}^{(0)} & a_{5,5}^{(1)} \end{bmatrix}.$$

In this way, we easily arrive at the following characterization of the lower triangular part of A_1.

Theorem 4.5. *We have*

$$\begin{aligned} \mathrm{tril}(A_1,0) &= \mathit{diag}(a_{1,1}^{(1)},\dots,a_{n,n}^{(1)}) + L(\{u_i\}_{i=2}^n,\{\overline{v_i}\}_{i=1}^{n-1},\{t_i\}_{i=2}^{n-1}) \\ &= \begin{bmatrix} a_{1,1}^{(1)} & 0 & \dots & \dots & 0 \\ u_2^{(1)}\overline{v_1^{(1)}} & a_{2,2}^{(1)} & \ddots & \ddots & \vdots \\ u_3^{(1)}t_2^{(1)}\overline{v_1^{(1)}} & u_3^{(1)}\overline{v_2^{(1)}} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ u_n^{(1)}t_{n-1}^{(1)}\dots t_2^{(1)}\overline{v_1^{(1)}} & u_n^{(1)}t_{n-1}^{(1)}\dots t_3^{(1)}\overline{v_2^{(1)}} & \dots & u_n^{(1)}\overline{v_{n-1}^{(1)}} & a_{n,n}^{(1)} \end{bmatrix}, \end{aligned}$$

where

$$u_j^{(1)} = \widehat{r}_{j,j}^{(0)},\ 2 \le j \le n, \quad t_j^{(1)} = \overline{\psi_{n-j}},\ 2 \le j \le n-1.$$

Moreover, the entries $a_{j,j}^{(1)}$ and $v_j^{(1)}$ are defined by the following procedure:

Set $v_1^{(1)} = \psi_{n-1}$, $a_{1,1}^{(1)} = \phi_{n-1}\widehat{r}_{1,1}^{(0)} + \overline{\psi}_{n-1}a_{1,2}^{(1/2)}$, and $\eta_2 = \phi_{n-1}$;
for $j = 2 : n-1$

1. $\overline{v_j^{(1)}} = \phi_{n-2+j}\overline{\psi_{n-j}}\eta_j + \overline{\psi_{n-2+j}}\phi_{n-j}$;
2. $a_{j,j}^{(1)} = \phi_{n-2+j}\eta_j\widehat{r}_{j,j}^{(0)} + \overline{\psi_{n-2+j}}a_{j,j+1}^{(1/2)}$;
3. $\eta_{j+1} = -\psi_{n+j-2}\overline{\psi_{n-j}}\eta_j + \phi_{n+j-2}\phi_{n-j}$.

end
Set $a_{n,n}^{(1)} = \eta_n\widehat{r}_{n,n}^{(0)}$.

This theorem says that the lower triangular part of A_1 and, therefore, the unknowns $d_1^{(1)}, \ldots, d_n^{(1)}$, $t_2^{(1)}, \ldots, t_{n-1}^{(1)}$, $\boldsymbol{u}^{(1)} \in \mathbb{C}^n$ and $\boldsymbol{v}^{(1)} \in \mathbb{C}^n$ can be evaluated at a linear cost whenever we know both the superdiagonal entries $a_{j,j+1}^{(1/2)}$ of $A_{1/2}$ and the elements $\widehat{r}_{j,j}^{(0)}$ emerging from the main diagonal in the construction of $A_{1/2}$.

Remark 4.6. Another interesting consequence of Theorem 4.5 is the minimality of the parametrization (1.2) used to represent the Hermitian matrices in $\mathcal{C}_n$. If $A = A_0 \in \mathcal{C}_n$ is Hermitian, then A_0 is a dpss matrix and this structure is preserved under the QR iteration, that is, A_s is dpss for any $s \geq 0$. At the very beginning, the representation (1.2) for A_0 involves $4n - 3$ parameters. The structure of the Hermitian dpss matrix A_1 obtained after one QR-step applied to A_0 is described in Theorem 4.5. By using the symmetry of A_1 it can easily be shown that the coefficients of the Givens rotations together with the elements $u_j^{(1)}$, $1 \leq j \leq n$, $(u_1^{(1)} = \widehat{r}_{1,1}^{(0)})$, are sufficient for representing the entries of A_1. In our formulas the cosine and sine appear separately because of numerical accuracy. Theoretically, only using the cosine (or sine) would be enough thus leading to a representation for A_1 involving $3n - 3$ parameters. The existence and the computation of such a minimal representation for any Hermitian dpss matrix is discussed in [56].

To compute these quantities efficiently, we devise suitable recurrence relations using the structural representation of R_0 provided by Theorem 4.3. The updating of $R_0^{(1)}$, i.e., the construction of the matrix $R_0^{(1)}(\mathcal{G}_{n-1,n}(\gamma_1))^H \ldots (\mathcal{G}_{2,3}(\gamma_{n-2}))^H$, can be carried out in a compact way at the cost of $O(n)$ ops by explicitly combining the columns of $R_0^{(1)}$ step by step. The updating of $R_0^{(2)}$ is also easily performed since the resulting matrix is lower triangular. Hence, we only need to specify the updating of $Q_0^{(1)} C_0$ and $Q_0^{(2)} C_0$. For the final updates of the latter matrices, we restrict ourselves to computing their diagonal and superdiagonal entries.

Recall from (4.10) that

$$C_0 = \begin{bmatrix} 0 & \overline{u_2^{(0)}} v_1^{(0)} & \overline{u_3^{(0)} t_2^{(0)}} v_1^{(0)} & \ldots & \overline{u_n^{(0)} t_{n-1}^{(0)} \ldots t_2^{(0)}} v_1^{(0)} \\ \vdots & \ddots & \overline{u_3^{(0)}} v_2^{(0)} & \ldots & \overline{u_n^{(0)} t_{n-1}^{(0)} \ldots t_3^{(0)}} v_2^{(0)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \overline{u_n^{(0)}} v_{n-1}^{(0)} \\ 0 & \ldots & \ldots & \ldots & 0 \end{bmatrix}$$

For demonstration consider the first two steps of the updating process applied to a 5×5 matrix C_0.

$$
C_0 \overset{\mathcal{G}_{4,5}(\gamma_1)^H}{\to} \begin{bmatrix} 0 & \times & \overline{u_3^{(0)} t_2^{(0)}} v_1^{(0)} & \tilde{u}_4^{(0)} \overline{t_3^{(0)} t_2^{(0)}} v_1^{(0)} & \widehat{u}_5^{(0)} \overline{t_3^{(0)} t_2^{(0)}} v_1^{(0)} \\ 0 & 0 & \overline{u_3^{(0)}} v_2^{(0)} & \tilde{u}_4^{(0)} \overline{t_3^{(0)}} v_2^{(0)} & \widehat{u}_5^{(0)} \overline{t_3^{(0)}} v_2^{(0)} \\ 0 & 0 & 0 & \tilde{u}_4^{(0)} v_3^{(0)} & \widehat{u}_5^{(0)} v_3^{(0)} \\ 0 & 0 & 0 & \eta_4 & \zeta_4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
\overset{\mathcal{G}_{3,4}(\gamma_2)^H}{\to} \begin{bmatrix} 0 & \overline{u_2^{(0)}} v_1^{(0)} & \tilde{u}_3^{(0)} \overline{t_2^{(0)}} v_1^{(0)} & \widehat{u}_4^{(0)} \overline{t_2^{(0)}} v_1^{(0)} & \times \\ 0 & 0 & \tilde{u}_3^{(0)} v_2^{(0)} & \widehat{u}_4^{(0)} v_2^{(0)} & \times \\ 0 & 0 & \eta_3 & \zeta_3 & \times \\ 0 & 0 & \overline{\psi_2} \eta_4 & \phi_2 \eta_4 & \times \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.17}
$$

where

$$
\begin{aligned}
\tilde{u}_4^{(0)} &= \phi_1 \overline{u_4^{(0)}} + \overline{\psi_1 u_5^{(0)} t_4^{(0)}}, \quad \tilde{u}_3^{(0)} = \phi_2 \overline{u_3^{(0)}} + \overline{\psi_2 \tilde{u}_4^{(0)} t_3^{(0)}}, \\
\widehat{u}_5^{(0)} &= -\psi_1 \overline{u_4^{(0)}} + \phi_1 \overline{u_5^{(0)} t_4^{(0)}}, \quad \widehat{u}_4^{(0)} = -\psi_2 \overline{u_3^{(0)}} + \phi_2 \overline{\tilde{u}_4^{(0)} t_3^{(0)}},
\end{aligned}
$$

and

$$
\begin{aligned}
\eta_4 &= \overline{\psi_1 u_5^{(0)}} v_4^{(0)}, \quad \eta_3 = \overline{\psi_2 \tilde{u}_4^{(0)}} v_3^{(0)}, \\
\zeta_4 &= \phi_1 \overline{u_5^{(0)}} v_4^{(0)}, \quad \zeta_3 = \phi_2 \overline{\tilde{u}_4^{(0)}} v_3^{(0)}.
\end{aligned}
$$

The updating of the 5×5 matrix C_0 is completed by

$$
C_0 \mathcal{G}_{4,5}(\gamma_1)^H \mathcal{G}_{3,4}(\gamma_2)^H \overset{\mathcal{G}_{2,3}(\gamma_3)^H}{\to} \begin{bmatrix} 0 & \tilde{u}_2^{(0)} v_1^{(0)} & \widehat{u}_3^{(0)} v_1^{(0)} & \times & \times \\ 0 & \eta_2 & \zeta_2 & \times & \times \\ 0 & \overline{\psi_3} \eta_3 & \phi_3 \eta_3 & \times & \times \\ 0 & \overline{\psi_3 \psi_2} \eta_4 & \phi_3 \overline{\psi_2} \eta_4 & \times & \times \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
$$

By proceeding in this way, we obtain the following description of $C_0(\mathcal{G}_{n-1,n}(\gamma_1))^H \ldots (\mathcal{G}_{j,j+1}(\gamma_{n-j}))^H$.

Theorem 4.7. *We have*

$$C_0(\mathcal{G}_{n-1,n}(\gamma_1))^H \dots (\mathcal{G}_{j,j+1}(\gamma_{n-j}))^H[1\colon n-1,\, j\colon n] =$$

$$\left[\begin{array}{c|ccc}
\tilde{u}_j^{(0)}\overline{t_{j-1}^{(0)}\dots t_2^{(0)}}v_1^{(0)} & \widehat{u}_{j+1}^{(0)}\overline{t_{j-1}^{(0)}\dots t_2^{(0)}}v_1^{(0)} & \dots & \dots & \widehat{u}_n^{(0)}\overline{t_{n-2}^{(0)}\dots t_2^{(0)}}v_1^{(0)} \\
\tilde{u}_j^{(0)}t_{j-1}^{(0)}\dots t_3^{(0)}v_1^{(0)} & \widehat{u}_{j+1}^{(0)}t_{j-1}^{(0)}\dots t_3^{(0)}v_1^{(0)} & \dots & \dots & \widehat{u}_n^{(0)}t_{n-2}^{(0)}\dots t_3^{(0)}v_1^{(0)} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\tilde{u}_j^{(0)}v_{j-1}^{(0)} & \widehat{u}_{j+1}^{(0)}v_{j-1}^{(0)} & \vdots & \vdots & \vdots \\
\hline
\eta_j & \zeta_j & \ddots & \vdots & \vdots \\
\overline{\psi_{n-j}}\eta_{j+1} & \phi_{n-j}\eta_{j+1} & \zeta_{j+1} & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \widehat{u}_n^{(0)}v_1^{(0)} \\
\overline{\psi_{n-j}\dots\psi_2}\eta_{n-1} & \overline{\psi_{n-j-1}\dots\psi_2}\phi_{n-j}\eta_{n-1} & \dots & \phi_2\eta_{n-1} & \zeta_{n-1}
\end{array}\right]$$

where

$$\begin{aligned}
\tilde{u}_{n-k}^{(0)} &= \phi_k\overline{u_{n-k}^{(0)}} + \overline{\psi_k}\tilde{u}_{n-k+1}^{(0)}\overline{t_{n-k}^{(0)}}, \quad 1 \le k \le n-j,\ (\tilde{u}_n^{(0)} = \overline{u_n^{(0)}})), \\
\widehat{u}_{n-k+1}^{(0)} &= -\psi_k\overline{u_{n-k}^{(0)}} + \phi_k\tilde{u}_{n-k+1}^{(0)}\overline{t_{n-k}^{(0)}}, \quad 1 \le k \le n-j,
\end{aligned}$$

and

$$\eta_{n-k} = \overline{\psi_k}\tilde{u}_{n-k+1}^{(0)}v_{n-k}^{(0)}, \quad \zeta_{n-k} = \phi_k\tilde{u}_{n-k+1}^{(0)}v_{n-k}^{(0)}, \quad 1 \le k \le n-j.$$

For $j = 2, \dots, n-1$, let $\widehat{c}_{j,j}^{(0)}$ and $\widehat{c}_{j,j+1}^{(0)}$ denote the entries of the matrix

$$(Q_0^{(1)} + Q_0^{(2)})C_0(\mathcal{G}_{n-1,n}(\gamma_1))^H \dots (\mathcal{G}_{j,j+1}(\gamma_{n-j}))^H$$

in positions (j, j) and $(j, j+1)$, respectively. The computation of these entries can be efficiently performed based on the recurrence relations which employ the quasiseparable structure of C_0, $Q_0^{(1)}$ and $Q_0^{(2)}$.

In view of Theorem 4.3, one easily finds that

$$\begin{aligned}
\widehat{c}_{j,j}^{(0)} = \tilde{u}_j^{(0)}\phi_{n-2+j}\Bigg[\mu_1\prod_{i=0}^{j-3}-\overline{\psi_{n+i}},\, \mu_2\prod_{i=1}^{j-3}-\overline{\psi_{n+i}},\, \dots,\, \mu_{j-1}\Bigg] \\
\times\begin{bmatrix}\overline{t_{j-1}^{(0)}\dots t_2^{(0)}}v_1^{(0)} \\ \vdots \\ v_{j-1}^{(0)}\end{bmatrix}
\end{aligned}$$

$$+\rho_j\eta_j + \delta_j\overline{\psi_{n-j}}\left[\phi_{n-j-1}, \phi_{n-j-2}\psi_{n-j-1}, \ldots, \tilde{\phi}\prod_{i=1}^{n-1-j}\psi_i\right]$$
$$\times\begin{bmatrix}\eta_{j+1}\\ \vdots\\ \eta_{n-1}\prod_{i=1}^{n-j-2}\overline{\psi_{n-j-i}}\\ 0\end{bmatrix},$$

and, similarly,

$$\widehat{c}^{(0)}_{j,j+1} = \widehat{u}^{(0)}_{j+1}\phi_{n-2+j}\left[\mu_1\prod_{i=0}^{j-3}-\overline{\psi_{n+i}}, \mu_2\prod_{i=1}^{j-3}-\overline{\psi_{n+i}}, \ldots, \mu_{j-1}\right]$$
$$\times\begin{bmatrix}\overline{t^{(0)}_{j-1}\ldots t^{(0)}_2}v^{(0)}_1\\ \vdots\\ v^{(0)}_{j-1}\end{bmatrix}$$
$$+\rho_j\zeta_j + \delta_j\phi_{n-j}\left[\phi_{n-j-1}, \phi_{n-j-2}\psi_{n-j-1}, \ldots, \tilde{\phi}\prod_{i=1}^{n-1-j}\psi_i\right]$$
$$\times\begin{bmatrix}\eta_{j+1}\\ \vdots\\ \eta_{n-1}\prod_{i=1}^{n-j-2}\overline{\psi_{n-j-i}}\\ 0\end{bmatrix}.$$

Thus, the computation of $\widehat{c}^{(0)}_{j,j}$ and $\widehat{c}^{(0)}_{j,j+1}$ reduces to evaluation of two scalar products:

$$s^{(0)}_j = \left[\mu_1\prod_{i=0}^{j-3}-\overline{\psi_{n+i}}, \mu_2\prod_{i=1}^{j-3}-\overline{\psi_{n+i}}, \ldots, \mu_{j-1}\right]\begin{bmatrix}\overline{t^{(0)}_{j-1}\ldots t^{(0)}_2}v^{(0)}_1\\ \vdots\\ v^{(0)}_{j-1}\end{bmatrix},$$

and

$$q^{(0)}_{n-j} = \left[\phi_{n-j-1}, \phi_{n-j-2}\psi_{n-j-1}, \ldots, \tilde{\phi}\prod_{i=1}^{n-1-j}\psi_i\right]\begin{bmatrix}\eta_{j+1}\\ \vdots\\ \eta_{n-1}\prod_{i=1}^{n-j-2}\overline{\psi_{n-j-i}}\\ 0\end{bmatrix}.$$

Observe that

$$s^{(0)}_j = (-\overline{\psi_{n-3+j}})\overline{t^{(0)}_{j-1}}s^{(0)}_{j-1} + \mu_{j-1}v^{(0)}_{j-1}, \quad 2 \le j \le n \tag{4.18}$$

and

$$(4.19)\quad q_{n-j}^{(0)} = |\psi_{n-j-1}|^2 q_{n-j-1}^{(0)} + \phi_{n-j-1}\eta_{j+1}, \quad j = n-2, \ldots, 1.$$

By using recursions (4.18) and (4.19) complemented with the initial conditions $s_1^{(0)} = 0$ and $q_1^{(0)} = 0$, we determine the entries $\widehat{c}_{j,j}^{(0)}$ and $\widehat{c}_{j,j+1}^{(0)}$ at a linear cost. Hence, the values $\widehat{r}_{j,j}^{(0)}$ as well as the superdiagonal entries of $A_{1/2}$ are also computed at a linear cost.

5 Implementation issues and numerical results

In this section we study the speed and the accuracy of the proposed algorithm. In particular, the structured QR iteration described in the previous sections has been implemented in MATLAB and then used for the computation of the eigenvalues of generalized companion matrices of both small and large size. The results of extensive numerical experiments confirm the robustness and the efficiency of the proposed approach.

At each QR step (1.1) the matrix $A_s \in \mathbb{C}^{n\times n}$ is stored in a linear data structure of size $\simeq 6n$ defined by the vectors $\boldsymbol{d}^{(s)}$, $\boldsymbol{u}^{(s)}$ $\boldsymbol{v}^{(s)}$, $\boldsymbol{z}^{(s)}$, $\boldsymbol{w}^{(s)}$ and $\boldsymbol{t}^{(s)}$. The QR step (1.1) is performed as follows.

function [$\boldsymbol{d}^{(1)}$, $\boldsymbol{u}^{(1)}$ $\boldsymbol{v}^{(1)}$, $\boldsymbol{z}^{(1)}$, $\boldsymbol{w}^{(1)}$, $\boldsymbol{t}^{(1)}$] = QRSStep($\boldsymbol{d}^{(0)}$, $\boldsymbol{u}^{(0)}$ $\boldsymbol{v}^{(0)}$, $\boldsymbol{z}^{(0)}$, $\boldsymbol{w}^{(0)}$ $\boldsymbol{t}^{(0)}$, σ_0)
% Compute the structured representation of A_1 generated by (1.1)
% A_1 is the matrix generated from A_0 after having performed one step
% of QR iteration with linear shift σ_0
Compute Givens rotations $\mathcal{G}_{n-1,n}(\gamma_1), \ldots, \mathcal{G}_{2,3}(\gamma_{n-2})$ by (4.5) and (4.6).
Set $a_{i,i}^{(0)} = d_i^{(0)} + z_i^{(0)}\overline{w_i^{(0)}} - \sigma_0$, $i = 1, \ldots, n$.
Find the generators of $\widehat{Q}_0 B_0$ as defined in Theorem 4.2.
Find Givens rotations $\mathcal{G}_{1,2}(\gamma_{n-1}), \ldots, \mathcal{G}_{n-1,n}(\gamma_{2n-3})$ as shown in Theorem 4.3.
Compute $\boldsymbol{z}^{(1)}$ and $\boldsymbol{w}^{(1)}$ as described in Remark 3.5.
Find the generators of $R_0^{(1)}$, $R_0^{(2)}$, $Q_0^{(1)}$ and $Q_0^{(2)}$ by means of Theorem 4.3.
Find $\boldsymbol{u}^{(1)}$ $\boldsymbol{v}^{(1)}$, $\boldsymbol{t}^{(1)}$ and $a_{i,i}^{(1)}$, $1 \le i \le n$, by using Theorem 4.5.
Evaluate $d_i^{(1)} = \mathrm{real}(a_{i,i}^{(1)} + \sigma_0 - z_i^{(1)}\overline{w_i^{(1)}})$, $i = 1, \ldots, n$.

Our implementation of `function QRSStep` requires $120\,n + O(1)$ multiplications and $28\,n + O(1)$ storage. The main program complements this routine with the following shifting strategy suggested in [[60], p. 549]. At the beginning the shift parameter σ is equal to zero. If $A_s = (a_{i,j}^{(s)}) \in \mathbb{C}^{n\times n}$ satisfies

$$(5.1)\qquad |a_{n,n}^{(s-1)} - a_{n,n}^{(s)}| \le 0.1|a_{n,n}^{(s-1)}|,$$

then we apply non-zero shifts by setting $\sigma_k = a_{n,n}^{(k)}$, $k = s, s+1, \ldots$. We say that $a_{n,n}^{(k)}$ provides a numerical approximation of an eigenvalue λ of A_0 whenever

$$|u_n^{(k)}| \max\{|v_{n-1}^{(k)}|, |t_{n-1}^{(k)}|\} \leq eps\ |a_{n,n}^{(k)}|,$$

where eps is the machine precision, i.e., $eps \simeq 2.2 \cdot 10^{-16}$. If this condition is fulfilled, then we set $\lambda = a_{n,n}^{(k)}$ and deflate the matrix by restarting the process with the initial matrix being the leading principal submatrix of A_k of order $n-1$.

After non-zero shifting has begun, we check for the convergence of the last diagonal entries of the currently computed iterate A_k. If convergence fails to occur after 15 iterations, then at the 16-th iteration we set $\sigma_k = 1.5\ (|a_{n,n}^{(k)}| + |u_n^{(k)} v_{n-1}^{(k)}|)$ and continue with non-zero shifting. If $a_{n,n}^{(k)}$ does not converge in the next 15 iterations, then the program reports failure. In our experience such failure has been never encountered.

The results of our numerical experiments are shown in Figures 1, 2, 3 and Table 1. Figure 1 covers our tests with unsymmetric arrowhead matrices A_0 obtained by setting $v_2^{(0)} = \ldots = v_n^{(0)} = 0$, $z_2^{(0)} = \ldots = z_n^{(0)} = 0$ and $t_2^{(0)} = \ldots = t_{n-1}^{(0)} = 1$, whereas the remaining elements $u_i^{(0)}$, $w_i^{(0)}$, $v_1^{(0)}$ and $z_1^{(0)}$ take random complex values with real and imaginary part ranging from -1 to 1 and $d_i^{(0)}$ are random real elements lying in the interval $[-1, 1]$. Table 1 covers the results of our tests with a set of arrowhead matrices of increasing order whose eigenvalues are algebraically known. Specifically, we consider the matrix $A_0 \in \mathbb{C}^{n \times n}$, $n = 2^s$, $s = 3, \ldots, 8$, having the unit entries on

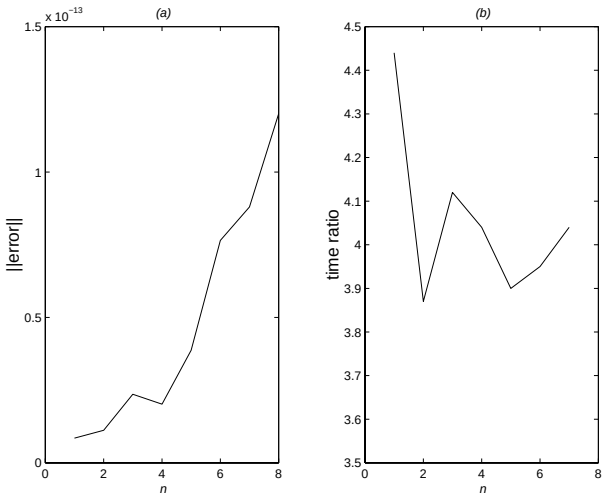

Fig. 1. Eigenvalue computation for unsymmetric arrowhead matrices of size $m(n) = 2^{2+n}$, $1 \leq n \leq 8$

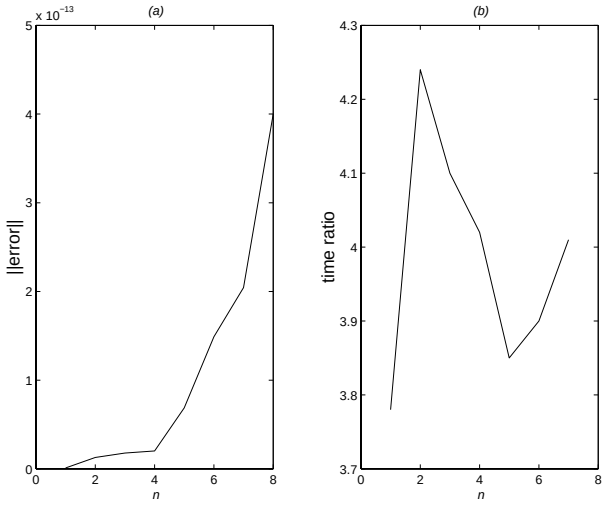

Fig. 2. Eigenvalue computation for Hermitian diagonal-plus-semiseparable matrices of size $m(n) = 2^{2+n}$, $1 \leq n \leq 8$

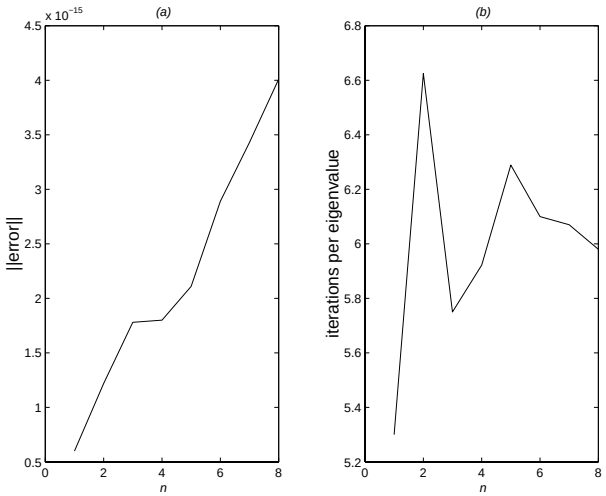

Fig. 3. Eigenvalue computation for Chebyshev comrade matrices of size $m(n) = 2^{2+n}$, $1 \leq n \leq 8$

the main diagonal and in the first row and the entries -1 in the first column (except for the position $(1, 1)$). Since $A_0^H = -A_0 + 2I_n$, we find that A_0 is normal and its eigenvalues are $1, 1-\sqrt{1-n}$ and $1+\sqrt{1-n}$. Figure 2 reports the test results for Hermitian diagonal-plus-semiseparable matrices, where $\boldsymbol{w}^{(0)} = \boldsymbol{z}^{(0)}$, $t_2^{(0)} = \ldots = t_{n-1}^{(0)} = 1$, $u_i^{(0)}$ and $v_i^{(0)}$ are random complex entries with real and imaginary parts ranging from -1 to 1 and $d_i^{(0)}$ are random real entries in the interval $[-1, 1]$. An implicit QR eigenvalue algorithm for

s	3	4	5	6	7	8
cond	1.2	1.5	3.8	2.4	7.2	1.2e+04
err	4.4e-16	1.4e-15	2.9e-15	6.7e-15	5.6e-14	1.5e-14

Table 1.

Hermitian dpss matrix is described in [54]. For Hermitian semiseparable matrices it requires $\simeq 41\,n$ flops per iteration. Our implementation does not exploit the symmetry of the input matrix and, therefore, its cost per iteration is about twice that of the algorithm in [54]. Figure 3 describes an application of our algorithm to polynomial root-finding. We consider the computation of the zeros of a complex n-degree polynomial given the coefficients of its representation in the Chebyshev basis defined by

$$\begin{cases} p_0(z) = 1 \\ p_j(z) = w^j + \left(\dfrac{1}{2w}\right)^j, \quad j \geq 1, \end{cases}$$

where $z = w + \dfrac{1}{2}w^{-1}$. In our experiments the coefficients are random complex values with real and imaginary parts ranging from -1 to 1. Since the Chebyshev basis satisfies a three-term recurrence relation, the polynomial root-finding problem can be recasted into a matrix setting as the problem of computing the eigenvalues of a Chebyshev-comrade matrix of order n.

Each figure contains two plots. In Figure 1 and 2 we show the values of the *errors* and running *time* for matrices A of size $m(n) = 2^{2+n}$ for $n = 1, \ldots, 8$. Figure 3 reports the values of the *errors* and average number of *iterations per eigenvalue* for matrices A of size $m(n) = 2^{2+n}$ for $n = 1, \ldots, 8$. Our test program returns these values as the output. The input data are random low rank perturbations of Hermitian matrices; experimentally we have found that their condition numbers with respect to eigenvalues are fairly small. Therefore, in our experiments we have always assumed that the MATLAB function `eig` computes the eigenvalues exactly. Let $\lambda(A)$ denote the set of eigenvalues computed by the MATLAB function `eig`. Let $\tilde{\lambda}(A)$ denote the set of eigenvalues computed by our algorithm, and define the distance between the sets $\lambda(A)$ and $\tilde{\lambda}(A)$ by

$$\mathrm{dist}(\lambda(A), \tilde{\lambda}(A)) = \max\{\max_{\tilde{\lambda}\in\tilde{\lambda}(A)} \parallel \tilde{\lambda} - \lambda(A) \parallel, \max_{\lambda\in\lambda(A)} \parallel \lambda - \tilde{\lambda}(A) \parallel\},$$

where $\parallel \lambda - \tilde{\lambda}(A) \parallel = \min_{\tilde{\lambda}\in\tilde{\lambda}(A)} |\lambda - \tilde{\lambda}|$. We refer to this distance as the error in the eigenvalues computed by our algorithm. For each size we carried out 100 numerical experiments. In each figure, the first plot reports the

average value of the errors. In Figure 1 and 2 the second plot reports the ratio between the average values of running time for matrices having sizes $m(n)$ and $m(n+1)$. Since $m(n+1)/m(n) = 2$ and the proposed algorithm for computing all the eigenvalues of A is expected to have quadratic cost, this ratio should be close to 4 for large $m(n)$. Table 1 reports the maximum estimated condition numbers and the distances between the set of the computed eigenvalues and the set of exact eigenvalues. These data are shown for the sizes $n = 2^s$, $s = 3, \ldots, 8$. Theoretically the condition number should be 1 due to the normality of A_0 but, in practice, the normality is lost after a few iterations so that the estimate provided by MATLAB can be far from 1.

The results of our numerical experiments are found to be quite accurate. This can be expected. Indeed, the proposed algorithm for computing all eigenvalues of a generalized companion matrix $A \in \mathcal{C}_n$ performs no division except in the computation of the Givens rotations. Furthermore, since it employs unitary transformations only, there is no coefficients growth at intermediate steps of the QR iteration. In particular, for the elements of the structured representation of the matrix $A_s = (a_{i,j}^{(s)}) \in \mathbb{C}^{n\times n}$ we have $\| \boldsymbol{z}^{(s)} \|_2 = \| \boldsymbol{z}^{(0)} \|_2$, $\| \boldsymbol{w}^{(s)} \|_2 = \| \boldsymbol{w}^{(0)} \|_2$, $\| \boldsymbol{t}^{(s)} \|_\infty \leq 1$, $\| \boldsymbol{v}^{(s)} \|_\infty \leq 1$, $\| \boldsymbol{u}^{(s)} \|_\infty \leq \| A_0 \|_2$ and $|a_{i,i}^{(s)}| \leq \| A_0 \|_2$, $1 \leq i \leq n$. (The result for $\boldsymbol{v}^{(s)}$ easily follows from Theorem 4.5 by using the Cauchy-Schwartz inequality to obtain an upper bound for its entries $v_j^{(s)}$, $1 \leq j \leq n$.) Therefore, the propagation of absolute errors during the iterative process can be taken under control, and this explains the robustness of the algorithm. In [7] the algorithm has been applied for approximating the roots of potentially ill-conditioned polynomials expressed in the Lagrange basis. In all the experiments reported there the accuracy of the computed approximations is in accordance with the conditioning estimates for the associated matrix eigenvalue problems.

The figures representing the running time confirm the effectiveness of our structured approach. According to our tests, the overall time for computing all the eigenvalues of A indeed changes roughly quadratically in n, in contrast to the classical QR method which requires $O(n^3)$ arithmetic operations and $O(n^2)$ storage.

Finally, from Figure 3 we deduce that the average number of iterations per eigenvalue is about 6.

6 Conclusion and future work

We have exploited matrix structures to devise a fast and numerically stable QR-based algorithm for computing all eigenvalues of a generalized companion matrix and, hence, all roots of associated polynomials and secular equations. This step has been highly desired but so far missing in the area of polynomial root-finding via matrix methods. The structural representation of

a generalized companion matrix based on its quasiseparable form is maintained at each step of the QR iteration, which enables us to yield a linear time per iteration using a linear memory space. Extensive numerical experiments have confirmed the effectiveness and the robustness of the proposed approach. We are advancing in a more refined implementation of the structured QR iteration including quadratic shifting techniques together with the optimization of the memory requirements. Another practical issue to be addressed is the selection of the cutting criterion for detecting whether the eigenvalue problem can be split into several smaller subproblems. The cheap cutting techniques introduced in [54] can easily be incorporated in our program. We plan to translate this program in Fortran to be compared with the LAPACK implementations of the QR algorithm.

Several extensions and applications of our results are now under investigation. The application to the polynomial root-finding problem causes no theoretical problem. Indeed, given a monic polynomial $p(z)$, a diagonal plus rank-one or arrowhead matrices A having $p(z)$ as their characteristic polynomial can be constructed at almost a linear cost. Our QR-like iteration applied to such matrices A provides an efficient polynomial root-finding algorithm with good convergence features. The theoretical properties as well as the numerical behavior of this QR-based polynomial root-finding algorithm will be described in a forthcoming paper.

In our approach the QR-step (1.1) is implemented explicitly and this is called the *explicit* QR algorithm. The *implicit* QR algorithm is based on a different way of carrying out the iteration (1.1). It is generally claimed [[59], p. 373] that the cost of the two single-shifted algorithms is almost the same, but in rare occasions the implicit version performs more stably. An implicit QR algorithm for the class of Hessenberg-like matrices, which includes generalized companion matrices as a special subclass, has been presented in [54]. It would be interesting to analyze the complexity of such an implicit QR algorithm applied to the computation of the eigenvalues of a generalized companion matrix $A \in \mathcal{C}_n$.

The problem of computing the set of eigenvectors of a generalized companion matrix given approximations of its eigenvalues also deserves further attention. It is well known that the inverse power iteration has serious drawbacks if clusters are present in the spectrum. A more refined method to compute the eigenvectors of a symmetric semiseparable matrix has been presented in [43]. The method is based on the properties of the implicitly shifted QR algorithm. Possible extensions to (symmetric) generalized companion matrices and to the explicit shifted QR algorithm are currently under study.

We have considered matrix classes which are small rank perturbations of Hermitian matrices. It turns out that simple fractional transformations convert an Hermitian matrix into a unitary matrix. This enables the extension of any method and result for almost Hermitian matrices to devising fast algorithms

for eigenvalue computation of small rank perturbations of unitary Hessenberg matrices. These generalizations and extensions will be presented elsewhere.

Frobenius (companion) matrices are most commonly used to compute polynomial roots. Frobenius matrices do not belong to the class of quasiseparable matrices introduced here. Notwithstanding that, by generalizing the approach of this paper it is possible to develop an algorithm for computing the eigenvalues of Frobenius matrices which uses $O(n)$ ops per iteration and a linear storage [5]. However, practical experience with this algorithm shows that numerical errors can sometimes magnify so that computed results are less accurate that the ones computed by the plain QR iteration. An alternative algorithm for Frobenius matrices relies on the extension extension of our methods to the eigenvalue computation of rank-one perturbations of unitary Hessenberg matrices. The design of a fast and stable algorithm for Frobenius matrices is therefore an ongoing work which still deserves further investigations.

Another interesting topic is the use of our results for the computation of all the eigenvalues of a general real matrix A. It is well known that the inverse of a nonsingular upper Hessenberg matrix takes the form $R + \boldsymbol{u}\boldsymbol{v}^T$, where R is an upper triangular matrix and $\boldsymbol{u}\boldsymbol{v}^T$ is a rank-one matrix. Furthermore, some recent algorithms for the numerical treatment of symmetric diagonal-plus-semiseparable matrices (see [55,57,58] and the references therein) transform A into a matrix of the same form $R + \boldsymbol{u}\boldsymbol{v}^T$, without using intermediate recurrence to a Hessenberg matrix. The transformation employs only unitary matrices so it is numerically robust. Once the reduction has been carried out, the matrix $R + \boldsymbol{u}\boldsymbol{v}^T$ with a nondefective R can be further transformed by a similarity transformation into a diagonal plus rank-one matrix B. The overall computational cost of computing B is $O(n^3)$ and the eigenvalues of B can be approximated by our algorithm at the total cost of $O(n^2)$. The second stage of the reduction from A to B, however, involves transformation matrices that generally are not unitary so that numerical difficulties could have arisen at this step. The design of a numerically stable algorithm that converts a real matrix A into a diagonal plus rank-one matrix by a similarity transformation as well as the experimental study of the numerical behavior of the latter algorithm are subjects of our work in progress.

Acknowledgements. The authors wish to thank the referees for their knowledgeable and helpful suggestions.

References

1. Arbenz, P., Golub, G.H.: QR-like algorithms for symmetric arrow matrices. SIAM J. Matrix Anal. Appl. **13**(2), 655–658 (1992)

2. Arbenz, P., Golub, G.H.: Matrix shapes invariant under the symmetric QR algorithm. Numer. Linear Algebra Appl. **2**, 87–93 (1995)
3. Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H.: Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide. Society for Industrial and Applied Mathematics (SIAM), 2000
4. Barnett, S.: A companion matrix analogue for orthogonal polynomials. Linear Algebra and Appl. **12**(3), 197–208 (1975)
5. Bini, D.A., Daddi, F., Gemignani, L.: On the shifted QR iteration applied to Frobenius matrices. To appear in ETNA
6. Bini, D.A., Gemignani, L., Pan, V.Y.: Inverse power and Durand-Kerner iterations for univariate polynomial root-finding. Comput. Math. with Appl. **47**, 447–459 (2004)
7. Bini, D.A., Gemignani, L., Pan, V.Y.: Improved initialization of the accelerated and robust QR-like polynomial root-finding. To appear in ETNA
8. Bini, D.A., Gemignani, L., Pan, V.Y.: QR-like algorithms for generalized semiseparable matrices. Rapporto Tecnico n. 1470, Dipartimento di Matematica, Università di Pisa, 2003
9. Bini, D., Gemignani, L.: Iteration schemes for the divide-and-conquer eigenvalue solver. Numer. Math. **67**, 403–425 (1994)
10. Bunch, J.R., Nielsen, C.P.: Updating the singular value decomposition. Numer. Math. **31**, 111–129 (1978)
11. Bunch, J.R., Nielsen, C.P., Sorensen, D.C.: Rank-one modification of the symmetric eigenproblem. Numer. Math. **31**, 31–48 (1978)
12. Calvetti, D., Kim, S., Reichel, L.: The restarted QR-algorithm for eigenvalue computation of structured matrices. J. Comput. Appl. Math. **149**, 415–422 (2002)
13. Carstensen, C.: Linear construction of companion matrices. Linear Algebra Appl. **149**, 191–214 (1991)
14. Chandrasekaran, S., Gu, M.: Fast and stable eigendecomposition of symmetric banded plus semiseparable matrices. Linear Algebra Appl. **313**, 107–114 (2000)
15. Chandrasekaran, S., Gu, M.: A fast and stable solver for recursively semi-separable systems of linear equations. In: Vadim Olshevsky (ed.), Structured Matrices in Mathematics, Computer, Science, and Engineering I, volume 281 of Contemporary Mathematics, American Mathematical Society (AMS), 2001, pp. 39–58
16. Chandrasekaran, S., Dewilde, P., Gu, M., Pals, T., van der Veen, A.J.: Fast stable solvers for sequentially semi-separable linear systems of equations. Lecture Notes in Computer Science **2552**, 545–554 (2002)
17. Cuppen, J.J.M.: A divide and conquer method for the symmetric tridiagonal eigenproblem. Numer. Math. **36**, 177–195 (1980/81)
18. Delvaux, S., Van Barel, M.: Rank structures preserved by the QR-algorithm: the singular case. Technical Report TW400, Katholieke Universiteit Leuven, Department Computerwetenschappen, 2004
19. Dewilde, P.: Systems of low Hankel rank: a survey. In: Vadim Olshevsky (ed.), Structured Matrices in Mathematics, Computer, Science, and Engineering I, volume 281 of Contemporary Mathematics, American Mathematical Society (AMS), 2001, pp. 91–102
20. Dewilde, P., van der Veen, A.J.: Time-varying systems and computations. Kluwer Academic Publishers, Boston, MA, 1998
21. Dongarra, J.J., Sorensen, D.C.: A fully parallel algorithm for the symmetric eigenvalue problem. SIAM J. Sci. Stat. Comput. **8**, 139–154 (1987)
22. Eidelman, Y., Gohberg, I:" Inversion formulas and linear complexity algorithm for diagonal plus semiseparable matrices. Comput. Math. Appl. **33**, 69–79 (1997)

23. Eidelman, Y., Gohberg, I.: On a new class of structured matrices. Integral Equations Operator Theory **34**, 293–324 (1999)
24. Eidelman, Y., Gohberg, I.: A modification of the Dewilde-van der Veen method for inversion of finite structured matrices. Linear Algebra Appl. **343–344**, 419–450 (2002)
25. Eidelman, Y., Gohberg, I.: Fast inversion algorithms for a class of structured operator matrices. Linear Algebra Appl. **371**, 153–190 (2003)
26. Elsner, L.: A remark on simultaneous inclusions of the zeros of a polynomial by Gershgorin's theorem. Numer. Math. **21**, 425–427 (1973)
27. Fadeeva, V.N.: Computational methods of Linear Algebra. Dover Publications, New York, NY, 1959
28. Fasino, D., Gemignani, L.: Structural and computational properties of possibly singular semiseparable matrices. Linear Algebra Appl. **340**, 183–198 (2002)
29. Fasino, D., Mastronardi, N., Van Barel, M.: Fast and stable algorithms for reducing diagonal plus semiseparable matrices to tridiagonal and bidiagonal form. In: Vadim Olshevsky (ed.), Fast Algorithms for Structured Matrices: Theory and Applications, volume 323 of Contemporary Mathematics, American Mathematical Society (AMS), 2003, pp. 105–118
30. Fiedler, M.: Expressing a polynomial as the characteristic polynomial of a symmetric matrix. Linear Algebra Appl. **141**, 265–270 (1990)
31. Fortune, S.: An iterated eigenvalue algorithm for approximating roots of univariate polynomials. J. Symbolic Comput. **33**(5), 627–646 (2002)
32. Fuhrmann, D.R.: An algorithm for subspace computation with applications in signal processing. SIAM J. Matrix Anal. Appl. **9**, 213–220 (1988)
33. Gander, W.: Least squares with a quadratic constraint. Numer. Math. **36**, 291–307 (1981)
34. Gander, W., Golub, G.H., von Matt, U.: A constrained eigenvalue problem. Linear Algebra Appl. **114–115**, 815–839 (1989)
35. Gantmacher, F.P., Krein, M.G.: Oscillation Matrices and Kernels and Small Vibrations of Mechanical Systems. Translation based on the 1941 Russian original, Edited and with a preface by Alex Eremenko, AMS Chelsea Publishing, Providence, RI, 2002
36. Gill, P.E., Golub, G.H., Murray, W., Saunders, M.A.: Methods for modifying matrix factorizations. Math. Comp. **28**, 505–535 (1974)
37. Golub, G.H.: Some modified matrix eigenvalue problems. SIAM Rev. **15**, 318–334 (1973)
38. Golub, G.H., Van Loan, C.F.: Matrix Computations. John Hopkins University Press, 1996
39. Golub, G.H., von Matt, U.: Quadratically constrained least squares and quadratic problems. Numer. Math. **59**, 561–580 (1991)
40. Gragg, W.B.: The QR algorithm for unitary Hessenberg matrices. J. Comput. Appl. Math. **16**, 1–8 (1986)
41. Koltracht, I.: Linear complexity algorithm for semiseparable matrices. Integral Equations Operator Theory **29**(3), 313–319 (1997)
42. Malek, F., Vaillancourt, R.: A composite polynomial zerofinding matrix algorithm. Comput. Math. Appl. **30**(2), 37–47 (1995)
43. Mastronardi, N., Van Barel, M., Vandebril, R.: On computing the eigenvectors of a class of structured matrices. In: Proceedings of the ICCAM 2004: Eleventh International Congress on Computational and Applied Mathematics, Katholieke Universiteit Leuven, Belgium, July, 2004
44. Melman, A.: Numerical solution of a secular equation. Numer. Math. **69**, 483–493 (1995)

45. Melman, A.: A unifying convergence analysis of second-order methods for secular equations. Math. Comp. **66**, 333–344 (1997)
46. Moler, C.: Cleve's Corner: ROOTS - Of polynomials, that is. The MathWorks Newsletter **5**(1), 8–9 (1991)
47. O'Leary, D.P., Stewart, G.W.: Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices. J. Comput. Phys. **90**(2), 497–505 (1990)
48. Parlett, B.N.: The symmetric eigenvalue problem. Society for Industrial and Applied Mathematics (SIAM), 1998
49. Solomonoff, A.: Application of multipole methods to two matrix eigenproblems. Preprint no. 1181, November 1993, IMA - Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, USA
50. Tyrtyshnikov, E.E.: Mosaic ranks for weakly semiseparable matrices. In: Large-scale scientific computations of engineering and environmental problems, II (Sozopol, 1999), volume **73** of Notes Numer. Fluid Mech. Vieweg, Braunschweig, 2000, pp. 36–41
51. Uhlig, F.: General polynomial roots and their multiplicities in $O(n)$ memory and $O(n^2)$ time. Linear and Multilinear Algebra **46**(4), 327–359 (1999)
52. Uhlig, F.: The DQR algorithm, basic theory, convergence, and conditional stability. Numer. Math. **76**(4), 515–553 (1997)
53. Van Barel, M., Fasino, D., Gemignani, L., Mastronardi, N.: Orthogonal rational functions. Technical Report TW350, Katholieke Universiteit Leuven, Department Computerwetenschappen, 2002. Submitted to SIAM J. Matrix Anal. Appl.
54. Vandebril, R.: Semiseparable matrices and the symmetric eigenvalue problem. PhD Thesis, Katholieke Universiteit Leuven, Department Computerwetenschappen, 2004
55. Vandebril, R., Van Barel, M., Mastronardi, N.: An orthogonal similarity reduction of a matrix to semiseparable form. Technical Report TW355, Katholieke Universiteit Leuven, Department Computerwetenschappen, 2003
56. Vandebril, R., Van Barel, M., Mastronardi, N.: A note on the representation and definition of semiseparable matrices. Technical Report TW368, Katholieke Universiteit Leuven, Department Computerwetenschappen, 2003
57. Vandebril, R., Van Barel, M., Mastronardi, N.: The Lanczos-Ritz values appearing in an orthogonal similarity reduction of a matrix into semiseparable form. Technical Report TW360, Katholieke Universiteit Leuven, Department Computerwetenschappen, 2003
58. Vandebril, R., Van Barel, M., Mastronardi, N.: A QR-method for computing the singular values via semiseparable. Technical Report TW366, Katholieke Universiteit Leuven, Department Computerwetenschappen, 2003
59. Watkins, D.S.: Fundamentals of matrix computations. Wiley-Interscience [John Wiley & Sons], New York, 2002
60. Wilkinson, J.H.: The algebraic eigenvalue problem. Clarendon Press, Oxford, 1965

this is the list of our corrections.

-- replace "ops" with "flops" at p.4, l.17; p.6, l.-13; p.23, l. 23; p.33, l.7

-- p.2 l. 14 replace "in" with "(specified in Subsection 1.3) by using"

-- p.5 l.20-21 replace "rank-one perturbations of unitary Hessenberg matrices (fellow matrices)" with "fellow matrices, that is, rank-one perturbations of unitary Hessenberg matrices,"

-- p.7 l. -18 replace "implied" with "implies"

--p.27 l.17-18 insert a comma after "u^{(1)}", "u^{(0)}" and "w^{(0)}"

--p.33 l. 11 remove "extension"

--pp. 34 update reference 5 and reference 7.

The reference 5 should be
"Bini, D. A., Daddi, F, Gemignani, L.: On the shifted QR iteration applied to companion matrices, Electron. Trans. Numer. Anal. {\bf 18}, 137-152 (2004)"

The reference 7 should be
"Bini, D. A., Gemignani, L., Pan, V.Y.: Improved initialization of the accelerated and robust QR-like polynomial root-finding, Electron. Trans. Numer. Anal. {\bf 17}, 195-205 (2004)"