Available online at www.sciencedirect.com

ScienceDirect

Indagationes Mathematicae 23 (2012) 733–761

indagationes mathematicae

www.elsevier.com/locate/indag

Implicit QR with compression

P. Boito[a], Y. Eidelman[c,*], L. Gemignani[b], I. Gohberg (Z"L)[d,1]

[a] *XLIM-DMI, UMR CNRS 7252, Faculté des Sciences et Techniques, Universit de Limoges, France*
[b] *Dipartimento di Matematica, Università di Pisa, Largo Bruno Pontecorvo 5, 56127 Pisa, Italy*
[c] *School of Mathematical Sciences, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel-Aviv University, Ramat-Aviv, 69978, Israel*
[d] *Department of Mathematics, Tel-Aviv University, Israel*

Abstract

In this paper, we elaborate on the implicit shifted QR eigenvalue algorithm given in [D.A. Bini, P. Boito, Y. Eidelman, L. Gemignani, I. Gohberg, A fast implicit QR eigenvalue algorithm for companion matrices, Linear Algebra Appl. 432 (2010), 2006–2031]. The algorithm is substantially simplified and speeded up while preserving its numerical robustness. This allows us to obtain a potentially important advance towards a proof of its backward stability together with both cost reductions and implementative benefits.

Keywords: Companion matrix; Polynomial roots; Quasiseparable matrices; QR method; Eigenvalue computation

When Israel Gohberg, the fourth author of this paper, passed away on October 12, 2009, the main results of the paper, especially the method of compression which lies in the basis of this work, were already obtained. The expression Z"L after his name is used in Hebrew and means "of blessed memory".

1. Introduction

The paper is motivated by the search of efficient and numerically robust methods for computing the complete set of eigenvalues of a companion matrix with application to the

* Corresponding author.
E-mail addresses: paola.boito@unilim.fr (P. Boito), eideyu@post.tau.ac.il (Y. Eidelman), gemignan@dm.unipi.it (L. Gemignani), gohberg@post.tau.ac.il (I. Gohberg (Z"L)).

[1] Deceased.

doi:10.1016/j.indag.2012.05.006

polynomial zero-finding problem. It is well known that for a Hermitian tridiagonal matrix the QR eigenvalue algorithm computes all the eigenvalues at a quadratic cost instead of the cubic cost required for dense Hessenberg matrices. Recent efforts have reached similar speed-up for certain classes of rank-structured matrices including small rank corrections of Hermitian and unitary matrices (see for a short summary [5] and the references given therein). The study of companion matrices started with the paper [2] by showing that the shifted QR iteration applied to a companion matrix $A \in \mathbb{C}^{N\times N}$ maintains the low rank structure of A. From then, a lot of algorithms have been developed that differ on the way used to parametrize and represent the rank structure. An up-to-date survey of these developments and algorithms can be found in the monograph [8].

The choice of the parametrization affects the computational and numerical properties of the resulting eigensolver. From one hand, for accuracy reasons it is important to control the growth of the parameters involved in the QR process by preserving at the same time the unitary plus rank-one structure of the initial companion matrix. These arguments strongly push in favor of the use of unitary-based parametrizations, where most of the coefficients are deduced from the representation of the rank structure in terms of unitary matrices. On the other hand, these parametrizations demand heavy computational effort, resulting in an increase of the big-O constant, and somewhat mask the original QR method, especially because the matrices generated under the process are only implicitly specified in terms of certain additional factors.

One way to alleviate this dichotomy is exploited in the implicit shifted QR eigenvalue algorithm for companion matrices described in our previous work [1]. That algorithm makes use of two different representations for specifying the matrices A_k, $k \geq 0$, $A_0 = A$ generated under the QR iteration and for carrying out each QR step $A_k \to A_{k+1}$. The composite scheme can be summarized as follows. The matrix A_k is initially provided as a rank-one perturbation of a unitary matrix given as product of two banded unitary matrices. These unitary factors are stored in factored form by the product of a linear number of 2×2 and 3×3 unitary matrices. Then, from the factorization of the unitary term an explicit entry-wise representation via generators is computed for the cumulative matrix A_k. This representation is employed to perform one step of the implicit shifted QR eigenvalue algorithm $A_k \to A_{k+1}$, where $A_{k+1} = \mathcal{G}_{N-1} \cdots \mathcal{G}_1 \cdot A_k \cdot \mathcal{G}_1^H \cdots \mathcal{G}_{N-1}^H$ and $\mathcal{G}_1, \ldots, \mathcal{G}_{N-1}$ are the unitary transformations generated by the implicit shifted QR iteration. Finally these transformations are applied to the matrix A_k represented in the initial unitary plus rank-one format to obtain the novel subsequent iterate.

Although the numerical experience reported in [1] is satisfactory in terms of accuracy and timings, it would be noted that the proposed approach can be prone to some drawbacks. In particular, the mechanism using two diverse representations is mathematically unsound, it makes difficult to prove theoretical stability results and, even more important, it is computationally burdensome to replicate for possible generalizations and extensions involving both companion pencils, matrix polynomials and block companion matrices. The main contribution of this paper is a substantial improvement of the implicit shifted QR eigenvalue algorithm presented in [1] aimed to circumvent all these issues.

More specifically, in this paper the structured QR iteration is greatly simplified by adopting a unique generator-based representation for all the matrices involved in the QR step. A technique named *compression* is introduced which makes it possible to compute the generators of the novel iterate A_{k+1} given the generators of the actual matrix A_k together with the transformations (Givens rotation matrices) generated by the implicit shifted QR scheme and with preservation of small orders of generators. The use of a unique parametrization is a potentially important advance towards the proof of the backward stability of the method. The compression process employs unitary matrices and yields a set of generators that are parts of unitary matrices so that the stability properties of the method in [1] are preserved or even enhanced. In addition, the resulting strategy has several computational benefits as it is cheaper, much simpler for implementation and easy to adjust for the block and the pencil case. The results of extensive numerical experiments are reported to confirm the practical impact and significance of these achievements.

The paper is organized as follows. In Section 2 we recall the structural properties and introduce condensed representations for the matrices generated by the QR process applied to an input companion matrix. In Section 3 we introduce the compression technique. Fast algorithms using this technique that carry out both the single-shift and the double-shift implicit QR iteration are described in Sections 4 and 5. In Section 6 the results of extensive numerical experiments are reported and, finally, the conclusion and a discussion of future work are the subjects of Section 7.

2. Definitions and properties of generators

For a given monic polynomial $\lambda(z)$ of degree N, $\lambda(z) = \sum_{i=0}^{N-1} a\lambda_i z^i + z^N$, the associated companion matrix $A \in \mathbb{C}^{N\times N}$ is

$$A = \begin{pmatrix} 0 & 0 & \cdots & 0 & -\lambda_0 \\ 1 & 0 & \cdots & 0 & -\lambda_1 \\ 0 & 0 & \cdots & 0 & -\lambda_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -\lambda_{N-1} \end{pmatrix}. \tag{2.1}$$

This paper is concerned with the problem of computing all the eigenvalues of A by means of the implicit shifted QR method

$$\begin{aligned} &A_0 = A \\ &q_k(A^{(k)}) = Q^{(k)} R^{(k)}, \quad \text{(QR factorization)} \\ &A^{(k+1)} := Q^{(k)*} A^{(k)} Q^{(k)}, \end{aligned} \tag{2.2}$$

where $q_k(z)$ is a monic polynomial of degree one (*single-shift step*) or two (*double-shift step*) suitably chosen to accelerate the convergence.

In order to analyze the structural properties of the matrices $A_k,\ k \geq 0$, it is useful to embed $A = A_0$ into a larger set. We denote by $\mathcal{H}_N$ the class of upper Hessenberg matrices $A \in \mathbb{C}^{N\times N}$ which are rank one perturbations of unitary matrices, i.e.,

$$A = U - pq^T, \tag{2.3}$$

where $U \in \mathbb{C}^{N\times N}$ is unitary and $p, q \in \mathbb{C}^N$. The vectors $p = (p(i))_{i=1}^N,\ q = (q(i))_{i=1}^N$ are called *the vectors of perturbation* for the matrix A.

The class $\mathcal{H}_N$ contains companion matrices of order N. In fact in the representation (2.3) for the companion matrix (2.1) the unitary matrix U can be taken as the circulant

$$U = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{2.4}$$

and the vectors of perturbation are

$$p = \begin{pmatrix} 1+\lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{n-1} \end{pmatrix}, \qquad q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}. \tag{2.5}$$

Since a matrix $A \in \mathcal{H}_N$ is upper Hessenberg, from (2.3) it follows that the entries below the first subdiagonal of the matrix U have the form

$$U(i,j) = p(i)q(j), \quad 1 \le j \le i-2,\ 3 \le i \le N. \tag{2.6}$$

We define the class $\mathcal{U}_N$ to be the set of $N \times N$ unitary matrices with the elements located in the lower triangular portion specified by $i - j \ge 2$, $1 \le i, j \le N$, of the form (2.6). The numbers $p(i)$ $(i = 3, \dots, N)$, $q(j)$ $(j = 1, \dots, N-2)$ are called *lower generators* of the matrix U. Notice that (2.6) implies

$$U(k, 1:k-2) = p(k)Q_{k-2}, \quad k = 3, \dots, N \tag{2.7}$$

with

$$Q_j = \begin{pmatrix} q(1) & \cdots & q(j) \end{pmatrix}, \quad j = 1, \dots, N-2. \tag{2.8}$$

We write

$$Q_j = \mathrm{row}(q(i))_{i=1}^{j}, \quad j = 1, \dots, N-2.$$

In the sequel we use the notation $\mathrm{row}(\cdot)$ to denote the matrix formed by appending column vectors of the same size, for instance, the numbers as above. Similarly, we use the notation $\mathrm{col}(\cdot)$ for the matrix obtained by stacking row vectors.

Assume that $U \in \mathbb{C}^{N \times N}$ has entries in the upper triangular part represented in the form

$$U(i,j) = g(i)b^{<}_{i-1,j}h(j), \quad 1 \le i \le j \le N, \tag{2.9}$$

where $b^{<}_{i-1,j} = b(i)\cdots b(j-1)$ for $i < j$ and $b^{<}_{i-1,i} = I_{r_i}$ with matrices $g(i), h(i)$ $(i = 1, \dots, N)$, $b(k)$ $(k = 1, \dots, N-1)$ of sizes $1 \times r_i, r_i \times 1, r_k \times r_{k+1}$ respectively. The elements $g(i), h(i)$ $(i = 1, \dots, N)$, $b(k)$ $(k = 1, \dots, N-1)$ are called *upper triangular generators* of the matrix U with orders r_k $(k = 1, \dots, N)$.

Every matrix U from the class $\mathcal{U}_N$ has upper triangular generators with orders not greater than two (see [1] for the proof). Hence, every matrix A from the class $\mathcal{H}_N$ defined in (2.3) is completely specified by the following parameters:

1. upper triangular generators $g(i), h(i)$ $(i = 1, \dots, N)$, $b(k)$ $(k = 1, \dots, N-1)$ of the unitary matrix U;
2. subdiagonal entries σ_k $(k = 1, \dots, N-1)$ of the matrix U;
3. the vectors of perturbation $p = (p(i))_{i=1}^{N}$, $q = (q(i))_{i=1}^{N}$.

For the companion matrix A in (2.1) the subdiagonal entries are $\sigma_k = 1$, $k = 1, \dots, N-1$, the vectors of perturbation are defined in (2.5) and upper triangular generators of the unitary matrix U from (2.4) are

$$\begin{aligned}&g(1)=1, \qquad g(i)=0, \quad i=2,\dots,N, \qquad h(j)=0, \quad j=1,\dots,N-1,\\ &h(N)=1,\\ &b(k)=1, \quad k=1,\dots,N-1.\end{aligned}$$

Upper triangular generators of a matrix are however not unique. Thus we conclude by mentioning some basic properties of generators which will be used in the next section to derive a minimal set of upper generators of a matrix $U \in \mathcal{U}_N$ and, a fortiori, of a matrix $A \in \mathcal{H}_N$.

Lemma 2.1. *Let $U \in \mathbb{C}^{N\times N}$ be a matrix with upper triangular generators $g(i), h(i)$ $(i=1,\dots,N)$, $b(k)$ $(k=1,\dots,N-1)$ with orders r_k $(k=1,\dots,N)$. Using these generators define the matrices G_k, H_k of sizes $k\times r_k, r_k\times(N-k+1)$, respectively, via the recursive relations*

$$G_1=g(1), \qquad G_k=\begin{pmatrix}G_{k-1}b(k-1)\\ g(k)\end{pmatrix}, \quad k=2,\dots,N \tag{2.10}$$

and

$$H_N=h(N), \qquad H_k=\begin{pmatrix}h(k) & b(k)H_{k+1}\end{pmatrix}, \quad k=N-1,\dots,1. \tag{2.11}$$

Then the relations

$$U(i,i:N)=g(i)H_i, \quad i=1,\dots,N, \tag{2.12}$$

and

$$U(1:i,i)=G_ih(i), \quad i=1,\dots,N \tag{2.13}$$

hold.

Proof. The recursions (2.12) and (2.13) mean

$$G_i=\mathrm{col}(g(k)b^{<}_{k-1,i})_{k=1}^{i}, \quad i=1,\dots,N \tag{2.14}$$

and

$$H_j=\mathrm{row}(b^{<}_{k,j+1}h(k))_{k=j}^{N}. \tag{2.15}$$

By comparing (2.14) and (2.15) with (2.9) we obtain (2.12) and (2.13). □

The reverse statement is also true. The proof immediately follows from (2.14) and (2.13).

Lemma 2.2. *Let $U \in \mathbb{C}^{N\times N}$ be matrix satisfying the relations (2.13), where G_k $(k=1,\dots,N)$ are matrices defined via the relations (2.10) for some matrices $g(i), h(i)$ $(i=1,\dots,N)$, $b(k)$ $(k=2,\dots,N-1)$. Then the elements $g(i), h(i)$ $(i=1,\dots,N)$, $b(k)$ $(k=1,\dots,N-1)$ are upper triangular generators of the matrix U.*

The next result describes a procedure which can be used to modify a set of generators by computing a possibly different set of generators of lower orders.

Lemma 2.3. *Let $U \in \mathbb{C}^{N\times N}$ be a matrix with upper triangular generators $g(i), h(i)$ $(i=1,\dots,N)$, $b(k)$ $(k=2,\dots,N-1)$ of orders r_k $(k=1,\dots,N)$. By using the generators $g(k), b(k)$ define the matrices G_k $(k=1,\dots,N-1)$ via relations (2.10). Suppose that for matrices*

S_k $(k = 1, \ldots, N)$ *of sizes* $r_k \times t_k$ *and matrices* $g^{(1)}(i), h^{(1)}(i)$ $(i = 1, \ldots, N)$, $b^{(1)}(k)$ $(k = 1, \ldots, N-1)$ *with corresponding sizes the relations*

$$g^{(1)}(k) = g(k)S_k, \quad k = 1, \ldots, N, \tag{2.16}$$

$$G_k S_k b^{(1)}(k) = G_k b(k) S_{k+1}, \quad k = 1, \ldots, N-1, \tag{2.17}$$

$$G_k S_k h^{(1)}(k) = G_k h(k), \quad k = 1, \ldots, N \tag{2.18}$$

are satisfied. Then the elements $g^{(1)}(i), h^{(1)}(i)$ $(i = 1, \ldots, N)$, $b^{(1)}(k)$ $(k = 1, \ldots, N-1)$ *are upper triangular generators of the matrix* U *of the orders* t_k $(k = 1, \ldots, N)$.

Proof. Define the matrices $G_k^{(1)}$ of sizes $k \times t_k$ via the recursive relations

$$G_1^{(1)} = g^{(1)}(1), \qquad G_k^{(1)} = \begin{pmatrix} G_{k-1}^{(1)} b^{(1)}(k-1) \\ g^{(1)}(k) \end{pmatrix}, \quad k = 2, \ldots, N. \tag{2.19}$$

We prove by induction the relations

$$G_k^{(1)} = G_k S_k, \quad k = 1, \ldots, N. \tag{2.20}$$

For $k = 1$ from (2.19), (2.10) and (2.16) we get

$$G_1^{(1)} = g^{(1)}(1) = g(1)S_1 = G_1 S_1.$$

Let us now assume that for some k with $1 \le k \le N-1$ the relation (2.20) holds. From (2.19), (2.16), (2.17) and (2.10) we find that

$$\begin{aligned} G_{k+1}^{(1)} &= \begin{pmatrix} G_k^{(1)} b^{(1)}(k) \\ g^{(1)}(k+1) \end{pmatrix} \\ &= \begin{pmatrix} G_k S_k b^{(1)}(k) \\ g(k+1) S_{k+1} \end{pmatrix} = \begin{pmatrix} G_k b(k) S_{k+1} \\ g(k+1) S_{k+1} \end{pmatrix} = G_{k+1} S_{k+1}, \end{aligned}$$

which completes the proof of (2.20). By using (2.18) and (2.20) we obtain

$$G_k^{(1)} h^{(1)}(k) = G_k S_k h^{(1)}(k) = G_k h(k), \quad k = 1, \ldots, N.$$

Hence by Lemma 2.1 it follows that

$$U(1:k, k) = G_k^{(1)} h^{(1)}(k), \quad k = 1, \ldots, N$$

and by Lemma 2.2 we conclude that $g^{(1)}(i), h^{(1)}(i)$ $(i = 1, \ldots, N)$, $b^{(1)}(k)$ $(k = 1, \ldots, N-1)$ are upper triangular generators of the matrix U. $\square$

3. The compression technique

Based on Lemma 2.3, in this section we introduce a novel method, referred to as the compression technique, that for any matrix from the class $\mathcal{U}_N$ with given upper triangular generators computes another set of generators with minimal orders. This method is at the core of the fast QR eigenvalue algorithm for companion matrices developed in the next two sections.

Theorem 3.1. *Let* $U \in \mathbb{C}^{N \times N}$ *be a unitary matrix from the class* $\mathcal{U}_N$ *with lower generators* $p(i)$ $(i = 3, \ldots, N)$, $q(j)$ $(j = 1, \ldots, N-2)$, *subdiagonal entries* σ_k $(k = 1, \ldots, N-1)$, *upper triangular generators* $g(i), h(i)$ $(i = 1, \ldots, N)$, $b(k)$ $(k = 1, \ldots, N-1)$ *of orders* r_k $(k = 1, \ldots, N)$. *Then a set of upper triangular generators* $g^{(1)}(i), h^{(1)}(i)$ $(i = 1, \ldots, N)$, $b^{(1)}(k)$ $(k = 1, \ldots, N-1)$ *of the matrix* U *of orders*

$$t_k = 2, \quad k = 1, \ldots, N-1, \qquad t_N = 1 \tag{3.1}$$

are obtained by using the following algorithm.

1. *Set*

$$S_N = h(N), \qquad h^{(1)}(N) = 1, \qquad g^{(1)}(N) = g(N)S_N \tag{3.2}$$

and

$$h^{(1)}(N-1) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad b^{(1)}(N-1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{3.3}$$

$$S_{N-1} = \begin{pmatrix} h(N-1) & b(N-1)S_N \end{pmatrix}, \tag{3.4}$$

$$g^{(1)}(N-1) = g(N-1)S_{N-1}, \tag{3.5}$$

$$z_N = \begin{pmatrix} \sigma_{N-1} & g^{(1)}(N) \end{pmatrix}, \qquad x_N = p(N). \tag{3.6}$$

2. *For* $k = N-1, \ldots, 2$ *perform the following.*
   (a) *Determine the complex Givens rotation* V_k *and the number* x_k *such that*

$$\begin{pmatrix} p(k) \\ x_{k+1} \end{pmatrix} = V_k \begin{pmatrix} x_k \\ 0 \end{pmatrix}. \tag{3.7}$$

   (b) *Compute*

$$\begin{pmatrix} a_k & y_k \\ f_k & d_k \end{pmatrix} = V_k^* \begin{pmatrix} \sigma_{k-1} & g^{(1)}(k) \\ x_{k+1}q(k-1) & z_{k+1} \end{pmatrix} \tag{3.8}$$

   with the matrices $a_k, y_k, f_k, d_k, z_{k+1}$ *of sizes* $1 \times 1, 1 \times 2, 1 \times 1, 1 \times 2, 1 \times 2$ *respectively. It is shown that the vector row* $\begin{pmatrix} f_k & d_k \end{pmatrix}$ *has unit norm so that one can determine a* 3×3 *unitary matrix* F_{k-1} *satisfying the condition*

$$\begin{pmatrix} f_k & d_k \end{pmatrix} F_{k-1}^* = \begin{pmatrix} 0_{1\times 2} & 1 \end{pmatrix}. \tag{3.9}$$

   (c) *Determine the matrices* $h^{(1)}(k-1), b^{(1)}(k-1)$ *of sizes* $2 \times 1, 2 \times 2$ *from the partition*

$$F_{k-1}(1:2,:) = \begin{pmatrix} h^{(1)}(k-1) & b^{(1)}(k-1) \end{pmatrix}. \tag{3.10}$$

   (d) *Compute the matrices* S_{k-1} *of the size* $r_{k-1} \times 2$ *and* z_k *of the size* 1×2 *by the formulas*

$$\begin{aligned} S_{k-1} &= h(k-1)(h^{(1)}(k-1))^* + b(k-1)S_k(b^{(1)}(k-1))^*, \\ z_k &= a_k(h^{(1)}(k-1))^* + y_k(b^{(1)}(k-1))^*. \end{aligned} \tag{3.11}$$

   (e) *Compute*

$$g^{(1)}(k-1) = g(k-1)S_{k-1}. \tag{3.12}$$

Proof. One should check that the relations (2.16)–(2.18) hold; hence, by Lemma 2.3 this implies that $g^{(1)}(i), h^{(1)}(i)$ $(i = 1, \ldots, N)$, and $b^{(1)}(k)$ $(k = 1, \ldots, N-1)$ are upper triangular generators of the matrix U. Moreover from the formulas (3.2), (3.3), (3.10), (3.12) it follows that the orders of these generators are in accordance with (3.1).

From (3.2) we obtain (2.18) and (2.16) with $k = N$. By using (3.3), (3.5), (3.4) we obtain (2.16)–(2.18) with $k = N-1$.

Next we prove by induction that all the matrices

$$\hat{U}_k = \begin{pmatrix} U(1:k,1:k-1) & G_k S_k \\ x_{k+1} Q_{k-1} & z_{k+1} \end{pmatrix}, \quad k = 2,\dots,N-1 \tag{3.13}$$

are unitary, the rows $\begin{pmatrix} f_k & d_k \end{pmatrix}$ $(k = N,\dots,2)$ have unit norms and the relations (2.16)–(2.18) hold.

Using (3.6) and (3.4) we have

$$\hat{U}_{N-1} = \begin{pmatrix} U(1:N-1,1:N-2) & G_{N-1}h(N-1) & G_{N-1}b(N-1)h(N) \\ p(N)Q_{N-2} & \sigma_{N-1} & g(N)h(N) \end{pmatrix}.$$

By using (2.7) with $k = N$, (2.13) with $i = N-1, N$ and (2.10) with $k = N$ we deduce that $\hat{U}_{N-1} = U$ and hence $\hat{U}_{N-1}$ is a unitary matrix. Suppose that for some k with $N-1 \geq k \geq 2$ the matrix $\hat{U}_k$ is unitary. By using (2.7), (2.13), (2.10) and the equality $Q_{k-1} = \begin{pmatrix} Q_{k-2} & q(k-1) \end{pmatrix}$ we get

$$\hat{U}_k = \begin{pmatrix} U(1:k-1,1:k-2) & G_{k-1}h(k-1) & G_{k-1}b(k-1)S_k \\ p(k)Q_{k-2} & \sigma_{k-1} & g(k)S_k \\ x_{k+1}Q_{k-2} & x_{k+1}q(k-1) & z_{k+1} \end{pmatrix}.$$

From the equalities (3.7), (3.8) and (3.5), (3.12) we obtain

$$\begin{pmatrix} I_{k-1} & 0 \\ 0 & V_k^* \end{pmatrix} \hat{U}_k = \begin{pmatrix} U(1:k-1,1:k-2) & G_{k-1}h(k-1) & G_{k-1}b(k-1)S_k \\ x_k Q_{k-2} & a_k & y_k \\ 0_{1\times(k-2)} & f_k & d_k \end{pmatrix}. \tag{3.14}$$

Since the matrix on the left hand side of (3.14) is unitary the three-dimensional row $\begin{pmatrix} f_k & d_k \end{pmatrix}$ has the unit norm. Hence one can determine a unitary 3×3 matrix F_{k-1} such that (3.9) holds. Further, from (3.10) we have

$$\begin{pmatrix} h(k-1) & b(k-1)S_k \\ a_k & y_k \end{pmatrix} F_{k-1}^* = \begin{pmatrix} S_{k-1} & w_k' \\ z_k & w_k'' \end{pmatrix} \tag{3.15}$$

with the matrices S_{k-1}, z_k of sizes $r_{k-1} \times 2$, 1×2 determined via (3.11) and some matrices w_k', w_k'' of sizes $r_{k-1} \times 1$, 1×1. Thus by using (3.14), (3.9) and (3.15) we get

$$\begin{pmatrix} I_{k-1} & 0 \\ 0 & V_k^* \end{pmatrix} \hat{U}_k \begin{pmatrix} I_{k-2} & 0 \\ 0 & F_{k-1}^* \end{pmatrix} = \begin{pmatrix} U(1:k-1,1:k-2) & G_{k-1}S_{k-1} & G_{k-1}w_k' \\ x_k Q_{k-2} & z_k & w_k'' \\ 0_{1\times(k-2)} & 0_{1\times 2} & 1 \end{pmatrix}. \tag{3.16}$$

Since the matrix on the left hand side of (3.16) is unitary we conclude that $G_{k-1}w_k' = 0,\ w_k'' = 0$ and, therefore,

$$\begin{pmatrix} I_{k-1} & 0 \\ 0 & V_k^* \end{pmatrix} \hat{U}_k \begin{pmatrix} I_{k-2} & 0 \\ 0 & F_{k-1}^* \end{pmatrix} = \begin{pmatrix} U(1:k-1,1:k-2) & G_{k-1}S_{k-1} & 0 \\ x_k Q_{k-2} & z_k & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \hat{U}_{k-1} & 0 \\ 0 & 1 \end{pmatrix}.$$

Hence it follows that the matrix $\hat{U}_{k-1}$ is unitary. Moreover from (3.15) it follows that

$$\begin{pmatrix} h(k-1) & b(k-1)S_k \end{pmatrix} F_{k-1}^* = \begin{pmatrix} S_{k-1} & w_k' \end{pmatrix}$$

and due to $G_{k-1}w_k' = 0$ we obtain

$$\begin{pmatrix} G_{k-1}h(k-1) & G_{k-1}b(k-1)S_k \end{pmatrix} = \begin{pmatrix} G_{k-1}S_{k-1} & 0 \end{pmatrix} F_{k-1}. \tag{3.17}$$

Finally, from (3.17) using the partition (3.10) we obtain the equalities (2.17), (2.18) with $k = N-2, \ldots, 1$. The relations (2.16) with $k = N-2, \ldots, 1$ follow directly from (3.12). □

It is worth noting that the representation of a matrix U from the class $\mathcal{U}_N$ by means of a set of upper generators of minimal orders constructed as in the proof of Theorem 3.1 can be related with a suitable factorization of U exploited for the design of the QR algorithm suggested in [1].

Remark 3.2. From the proof of Theorem 3.1 one can derive easily that every matrix U from the class $\mathcal{U}_N$ admits the factorization

$$U = V \cdot F,$$

where

$$V = \tilde{V}_{N-1} \cdots \tilde{V}_2, \qquad F = \tilde{F}_1 \cdots \tilde{F}_{N-2},$$
$$\tilde{V}_i = I_{i-1} \oplus V_i \oplus I_{N-i-1}, \qquad \tilde{F}_i = I_{i-1} \oplus F_i \oplus I_{N-i-2}$$

with 2×2 unitary matrices V_i and 3×3 unitary matrices F_i defined in the theorem.

4. The structured QR iteration: the single-shift case

Let $A \in \mathbb{C}^{N \times N}$ be an upper Hessenberg matrix. Let us consider one single-shift step of the QR iteration (2.2) for this matrix with shift polynomial $q(z) = z - \alpha$. The implicit QR algorithm consists of the computation of the unitary matrix Q of the form

$$Q = \tilde{Q}_1 \tilde{Q}_2 \cdots \tilde{Q}_{N-1}, \tag{4.1}$$

where

$$\tilde{Q}_i = I_{i-1} \oplus Q_i \oplus I_{N-i-1} \tag{4.2}$$

and Q_i, $i = 1, \ldots, N-1$ are complex Givens rotation matrices determined so that $(\tilde{Q}_1^* q(A))(2,1) = 0$ and, moreover, $A^{(1)} = Q^* A Q$ is upper Hessenberg.

The restoration of the Hessenberg form is performed by means of a bulge-chasing procedure. First we compute the matrices $A_1' = \tilde{Q}_1^* A$, $A_1 = A_1' \tilde{Q}_1$. Then we observe that the matrix A_1' is upper Hessenberg and the matrix A_1 is similar to the matrix A and contains a nonzero entry in the $(3,1)$ position. We choose the matrix Q_2 in order to annihilate this entry, i.e., to get $(\tilde{Q}_2^* A_1)(3,1) = 0$. Then we compute the matrices $A_2' = \tilde{Q}_2^* A_1$, $A_2 = A_2' \tilde{Q}_2$, the matrix A_2' is upper Hessenberg and the matrix A_2 is similar to the matrix A and contains a nonzero entry in the $(4,2)$ position. We choose the matrix Q_3 in order to annihilate this entry, i.e., to get $(\tilde{Q}_3^* A_2)(4,2) = 0$. Then we compute the matrices $A_3' = \tilde{Q}_3^* A_2$, $A_3 = A_3' \tilde{Q}_3$, the matrix A_3' is upper Hessenberg and the matrix A_3 is similar to the matrix A and contains a nonzero entry in the $(5,3)$ position. We continue this procedure and obtain the sequence of matrices

$$\begin{aligned} &A_0 = A; \qquad A_k' = \tilde{Q}_k^* A_{k-1}, \qquad A_k = A_k' \tilde{Q}_k, \quad k = 1, \ldots, N-1; \\ &A^{(1)} := A_{N-1}. \end{aligned} \tag{4.3}$$

Here all the matrices $A'_k,\ k = 1, \dots, N-1$, and the matrix $A^{(1)}$ are upper Hessenberg and all the matrices $A_k,\ k = 1, \dots, N-1$, are similar to the matrix A.

In the sequel of this section we present a fast adaptation of the implicit single-shift QR algorithm for an input matrix $A \in \mathcal{H}_N$. The modified algorithm works on the generators of the matrix and this explains why it is referred to as a structured QR iteration.

Using the decomposition (2.3) and setting

$$\begin{aligned} &U_0 = U; \qquad U'_k = \tilde{Q}_k^* U_{k-1}, \qquad U_k = U'_k \tilde{Q}_k, \quad k = 1, \dots, N-1; \\ &U^{(1)} := U_{N-1}, \end{aligned} \tag{4.4}$$

$$p_0 = p; \qquad p_k = \tilde{Q}_k^* p_{k-1}, \quad k = 1, \dots, N-1; \qquad p^{(1)} := p_{N-1} \tag{4.5}$$

$$q_0 = q; \qquad q_k = \tilde{Q}_k^T q_k, \quad k = 1, \dots, N-1; \qquad q^{(1)} := q_{N-1} \tag{4.6}$$

we get

$$A'_k = U'_k - p_k q_{k-1}^T, \qquad A_k = U_k - p_k q_k^T, \quad k = 1, \dots, N-1. \tag{4.7}$$

Here U_k, U'_k are unitary matrices and p_k, q_k are vectors. The new iterate $A^{(1)}$ has therefore the form

$$A^{(1)} = U^{(1)} - p^{(1)} (q^{(1)})^T. \tag{4.8}$$

The problem of computing a set of generators of the matrix $A^{(1)}$ is addressed in the following result.

Theorem 4.1. *Let $A = U - pq^T$ be a matrix from the class $\mathcal{H}_N$ with vectors of perturbation $p = \mathrm{col}(p(i))_{i=1}^N$, $q = \mathrm{col}(q(i))_{i=1}^N$, upper triangular generators $g(i), h(i)$ $(i = 1, \dots, N)$, $b(k)$ $(k = 1, \dots, N-1)$ of orders r_k $(k = 1, \dots, N)$ and subdiagonal entries σ_k $(k = 1, \dots, N-1)$ of the matrix U, and let α be a number. Then the Givens rotation matrices Q_k, $k = 1, \dots, N-1$, the vectors of perturbation $p^{(1)} = \mathrm{col}(p^{(1)}(i))_{i=1}^N$, $q^{(1)} = \mathrm{col}(q^{(1)}(i))_{i=1}^N$ of the upper Hessenberg matrix $A^{(1)} = Q^* A Q$ as well as the subdiagonal entries $\sigma_k^{(1)}$ $(k = 1, \dots, N-1)$ and the upper triangular generators $g^{(1)}(i), h^{(1)}(i)$ $(i = 1, \dots, N)$, $b^{(1)}(k)$ $(k = 1, \dots, N-1)$ with orders not greater than two of the unitary matrix $U^{(1)}$ in (4.8) are obtained by the following procedure.*

1. *Compute the Givens rotation matrices $Q_k,\ k = 1, \dots, N-1$, the vectors of perturbation $p^{(1)} = \mathrm{col}(p^{(1)}(i))_{i=1}^N$, $q^{(1)} = \mathrm{col}(q^{(1)}(i))_{i=1}^N$ of the upper Hessenberg matrix $A^{(1)} = Q^* A Q$ and the subdiagonal entries $\sigma_k^{(1)}$ $(k = 1, \dots, N-1)$ of the unitary matrix $U^{(1)}$, by performing these steps.*
   (a) *Determine the complex Givens rotation matrix Q_1 from the condition*

$$Q_1^* \begin{pmatrix} g(1)h(1) - p(1)q(1) - \alpha \\ \sigma_1 - p(2)q(1) \end{pmatrix} = \begin{pmatrix} * \\ 0 \end{pmatrix}. \tag{4.9}$$

Compute

$$\begin{pmatrix} \gamma_1 & w_1 \\ f_2 & v_2 \end{pmatrix} = Q_1^* \begin{pmatrix} g(1)h(1) & g(1)b(1) \\ \sigma_1 & g(2) \end{pmatrix}, \tag{4.10}$$

$$\begin{pmatrix} s_2 & \eta_2 \\ \rho_2 & \beta_2 \end{pmatrix} = \begin{pmatrix} f_2 & v_2 h(2) \\ p(3)q(1) & \sigma_2 \end{pmatrix} Q_1, \tag{4.11}$$

$$\begin{pmatrix} p^{(1)}(1) \\ c_2 \end{pmatrix} = Q_1^* \begin{pmatrix} p(1) \\ p(2) \end{pmatrix}, \qquad \begin{pmatrix} q^{(1)}(1) & \theta_2 \end{pmatrix} = \begin{pmatrix} q(1) & q(2) \end{pmatrix} Q_1 \tag{4.12}$$

with numbers $\gamma_1, f_2, s_2, \rho_2, \eta_2, \beta_2, c_2, \theta_2$ *and* r_2*-dimensional rows* w_1 *and* v_2*.*

(b) *For* $k = 2, \dots, N-2$ *perform the following computation. Determine the complex Givens rotation matrix* Q_k *and the number* δ_{k-1} *such that*

$$Q_k^* \begin{pmatrix} s_k - c_k q^{(1)}(k-1) \\ \rho_k - p(k+1)q^{(1)}(k-1) \end{pmatrix} = \begin{pmatrix} \delta_{k-1} \\ 0 \end{pmatrix}. \tag{4.13}$$

Compute

$$\begin{pmatrix} \gamma_k & w_k \\ f_{k+1} & v_{k+1} \end{pmatrix} = Q_k^* \begin{pmatrix} \eta_k & v_k b(k) \\ \beta_k & g(k+1) \end{pmatrix}, \tag{4.14}$$

$$\begin{pmatrix} s_{k+1} & \eta_{k+1} \\ \rho_{k+1} & \beta_{k+1} \end{pmatrix} = \begin{pmatrix} f_{k+1} & v_{k+1}h(k+1) \\ p(k+2)\theta_k & \sigma_{k+1} \end{pmatrix} Q_k, \tag{4.15}$$

$$\begin{gathered} \begin{pmatrix} p^{(1)}(k) \\ c_{k+1} \end{pmatrix} = Q_k^* \begin{pmatrix} c_k \\ p(k+1) \end{pmatrix}, \\ \begin{pmatrix} q^{(1)}(k) & \theta_{k+1} \end{pmatrix} = \begin{pmatrix} \theta_k & q(k+1) \end{pmatrix} Q_k, \end{gathered} \tag{4.16}$$

$$\sigma_{k-1}^{(1)} = \delta_{k-1} + p^{(1)}(k)q^{(1)}(k-1) \tag{4.17}$$

with numbers $\gamma_k, f_{k+1}, s_{k+1}, \rho_{k+1}, \eta_{k+1}, \beta_{k+1}, c_{k+1}, \theta_{k+1}$ *and* r_{k+1}*-dimensional rows* w_k, v_{k+1}*.*

(c) *Determine the complex Givens rotation matrix* Q_{N-1} *and the number* δ_{N-2} *such that*

$$Q_{N-1}^* \begin{pmatrix} s_{N-1} - c_{N-1}q^{(1)}(N-2) \\ \rho_{N-1} - p(N)q^{(1)}(N-2) \end{pmatrix} = \begin{pmatrix} \delta_{N-2} \\ 0 \end{pmatrix}. \tag{4.18}$$

Compute

$$\begin{pmatrix} \gamma_{N-1} & w_{N-1} \\ f_N & v_N \end{pmatrix} = Q_{N-1}^* \begin{pmatrix} \eta_{N-1} & v_{N-1}b(N-1) \\ \beta_{N-1} & g(N) \end{pmatrix}, \tag{4.19}$$

$$\begin{pmatrix} s_N & \eta_N \end{pmatrix} = \begin{pmatrix} f_N & v_N h(N) \end{pmatrix} Q_{N-1}, \tag{4.20}$$

$$\begin{gathered} \begin{pmatrix} p^{(1)}(N-1) \\ p^{(1)}(N) \end{pmatrix} = Q_{N-1}^* \begin{pmatrix} c_{N-1} \\ p(N) \end{pmatrix}, \\ \begin{pmatrix} q^{(1)}(N-1) & q^{(1)}(N) \end{pmatrix} = \begin{pmatrix} \theta_{N-1} & q(N) \end{pmatrix} Q_{N-1}, \end{gathered} \tag{4.21}$$

$$\sigma_{N-2}^{(1)} = \delta_{N-2} + p^{(1)}(N-1)q^{(1)}(N-2) \tag{4.22}$$

with numbers $\gamma_{N-1}, f_N, s_N, \eta_N$ *and* r_N*-dimensional rows* w_{N-1}, v_N*. Set* $\sigma_{N-1}^{(1)} = s_N$*.*

2. *Determine the numbers* $h_Q(k), b_Q(k), d_Q(k), g_Q(k)$ *from the partition*

$$Q_k = \begin{pmatrix} h_Q(k) & b_Q(k) \\ d_Q(k) & g_Q(k) \end{pmatrix}, \quad k = 1, \dots, N-1 \tag{4.23}$$

and set

$$g_U(k) = \begin{pmatrix} \gamma_k & w_k \end{pmatrix}, \quad k = 1, \dots, N-1, \qquad g_U(N) = \eta_N, \tag{4.24}$$

$$h_U(k) = \begin{pmatrix} h_Q(k) \\ h(k+1)d_Q(k) \end{pmatrix}, \quad k = 1, \dots, N-1, \qquad h_U(N) = 1, \tag{4.25}$$

$$\begin{aligned} b_U(k) &= \begin{pmatrix} b_Q(k) & 0 \\ h(k+1)g_Q(k) & b(k+1) \end{pmatrix}, \quad k = 1, \ldots, N-2, \\ b_U(N-1) &= \begin{pmatrix} b_Q(N-1) \\ h(N)g_Q(N-1) \end{pmatrix}. \end{aligned} \tag{4.26}$$

3. *By using the algorithm from Theorem 3.1 with lower generators $p^{(1)}(i)$ $(i = 3, \ldots, N)$, $q^{(1)}(j)$ $(j = 1, \ldots, N-2)$, subdiagonal entries $\sigma_k^{(1)}$ $(k = 1, \ldots, N-1)$ and upper triangular generators $g_U(i), h_U(i)$ $(i = 1, \ldots, N)$, and $b_U(k)$ $(k = 1, \ldots, N-1)$ with orders $r'_k = r_k + 1,\ k = 1, \ldots, N-1,\ r'_N = 1$ determine upper triangular generators $g^{(1)}(i), h^{(1)}(i)$ $(i = 1, \ldots, N)$, $b^{(1)}(k)$ $(k = 1, \ldots, N-1)$ of orders*

$$t_k = 2, \quad k = 1, \ldots, N-1, \qquad t_N = 1 \tag{4.27}$$

of the matrix $U^{(1)}$.

Proof. From (4.2) it follows that premultiplication of a matrix C by the matrix $\tilde{Q}_k^*$ means only premultiplication of the rows $k, k+1$ of C by the matrix Q_k^* and postmultiplication of a matrix C by the matrix $\tilde{Q}_k$ means only postmultiplication of the columns $k, k+1$ of C by the matrix Q_k. This in particular implies that the vectors p_k, q_k from (4.5), (4.6) have the form

$$p_k = (p^{(1)}(1), \ldots, p^{(1)}(k), c_{k+1}, p(k+2), \ldots, p(N))^T, \quad k = 1, \ldots, N-2, \tag{4.28}$$

$$q_k = (q^{(1)}(1), \ldots, q^{(1)}(k), \theta_{k+1}, q(k+2), \ldots, q(N))^T, \quad k = 1, \ldots, N-2 \tag{4.29}$$

and moreover the formulas (4.12), (4.16), (4.21) yield the coordinates of the vectors of perturbation $p^{(1)} = Q^* p,\ q^{(1)} = Q^T q$. Furthermore from (4.4) it follows that

$$U'_k(1:k, 1:k-1) = U^{(1)}(1:k, 1:k-1), \quad k = 2, \ldots, N-1, \tag{4.30}$$

$$U'_k(k+2:N, k+1:N) = U(k+2:N, k+1:N), \quad k = 1, \ldots, N-2 \tag{4.31}$$

and

$$U_k(1:k, 1:k) = U^{(1)}(1:k, 1:k), \quad k = 1, \ldots, N-1. \tag{4.32}$$

By using the matrices $g_U(k), b_U(k)$ define the matrices G_k^U via the recursive relations

$$G_1^U = g_U(1), \qquad G_k^U = \begin{pmatrix} G_{k-1}^U b_U(k-1) \\ g_U(k) \end{pmatrix}, \quad k = 2, \ldots, N. \tag{4.33}$$

From (4.24), (4.26) we have

$$\begin{aligned} G_1^U &= \begin{pmatrix} \gamma_1 & w_1 \end{pmatrix}, \qquad G_k^U = \begin{pmatrix} G_{k-1}^U \tilde{b}(k) & G_{k-1}^U \hat{b}(k) \\ \gamma_k & w_k \end{pmatrix}, \quad k = 2, \ldots, N-1, \\ G_N^U &= \begin{pmatrix} G_{N-1}^U b_U(N-1) \\ \eta_N \end{pmatrix} \end{aligned} \tag{4.34}$$

with

$$\tilde{b}(k) = \begin{pmatrix} b_Q(k-1) \\ h(k)g_Q(k-1) \end{pmatrix}, \qquad \hat{b}(k) = \begin{pmatrix} 0_{1\times r_{k+1}} \\ b(k) \end{pmatrix}, \quad k = 2, \ldots, N-1. \tag{4.35}$$

Notice that in the view of (4.23) one can rewrite the formulas (4.25), (4.26) for $k = 1, \dots, N-2$ in the form

$$\begin{pmatrix} h_U(k) & \tilde{b}(k+1) & \hat{b}(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & h(k+1) & b(k+1) \end{pmatrix} \begin{pmatrix} Q_k & 0 \\ 0 & I_{r_{k+2}} \end{pmatrix}, \tag{4.36}$$

and

$$\begin{pmatrix} h_U(N-1) & b_U(N-1) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & h(N) \end{pmatrix} Q_{N-1}. \tag{4.37}$$

We prove by induction that the Givens rotation matrices Q_k with $k = 1, \dots, N-2$ are determined via (4.9), (4.13) and, moreover, that the relations

$$U_k(1:k+2, k:N) = \begin{pmatrix} G_k^U h_U(k) & G_k^U \tilde{b}(k+1) & G_k^U \hat{b}(k+1) H_{k+2} \\ s_{k+1} & \eta_{k+1} & v_{k+1} b(k+1) H_{k+2} \\ \rho_{k+1} & \beta_{k+1} & g(k+2) H_{k+2} \end{pmatrix}, \tag{4.38}$$

hold for $k = 1, \dots, N-2$.

From (2.9) and (2.3) we obtain

$$(A - \alpha I)(1:2, 1) = \begin{pmatrix} g(1)h(1) - p(1)q(1) - \alpha \\ \sigma_1 - p(2)q(1) \end{pmatrix}$$

which yields the condition (4.9) for the matrix Q_1.

By using (2.6) with $i = 3$, $j = 1$, (2.12) with $i = 1, 2, 3$ and (2.11) with $k = 1, 2$ we find that

$$U(1:3, 1:N) = \begin{pmatrix} g(1)h(1) & g(1)b(1)h(2) & g(1)b(1)b(2)H_3 \\ \sigma_1 & g(2)h(2) & g(2)b(2)H_3 \\ p(3)q(1) & \sigma_2 & g(3)H_3 \end{pmatrix}. \tag{4.39}$$

From (4.10) we get

$$U_1'(1:3, 1:N) = \begin{pmatrix} \gamma_1 & w_1 h(2) & w_1 b(2) H_3 \\ f_2 & v_2 h(2) & v_2 b(2) H_3 \\ p(3)q(1) & \sigma_2 & g(3) H_3 \end{pmatrix}.$$

The first row of the matrix U_1' can be expressed in the form

$$U_1'(1, 1:N) = \begin{pmatrix} \gamma_1 & w_1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & h(2) & b(2) \end{pmatrix} \begin{pmatrix} I_2 & 0 \\ 0 & H_3 \end{pmatrix}. \tag{4.40}$$

From (4.34) and (4.36) with $k = 1$ we have

$$\begin{aligned} U_1(1, 1:N) &= G_1^U \begin{pmatrix} h_U(1) & \tilde{b}(2) & \hat{b}(2) \end{pmatrix} \begin{pmatrix} I_2 & 0 \\ 0 & H_3 \end{pmatrix} \\ &= \begin{pmatrix} G_1^U h_U(1) & G_1^U \tilde{b}(2) & G_1^U \hat{b}(2) H_3 \end{pmatrix}. \end{aligned} \tag{4.41}$$

By using (4.11) we find that

$$U_1(2:3, 1:N) = \begin{pmatrix} s_2 & \eta_2 & v_2 b(2) H_3 \\ \rho_2 & \beta_2 & g(3) H_3 \end{pmatrix}. \tag{4.42}$$

By combining (4.41) and (4.42) together we obtain (4.38) with $k = 1$.

Let us now assume that for some k with $2 \le k \le N-3$ the representation

$$U_{k-1}(1:k+1,k-1:N) = \begin{pmatrix} G^U_{k-1}h_U(k-1) & G^U_{k-1}\tilde{b}(k) & G^U_{k-1}\hat{b}(k)H_{k+1} \\ s_k & \eta_k & v_k b(k)H_{k+1} \\ \rho_k & \beta_k & g(k+1)H_{k+1} \end{pmatrix} \tag{4.43}$$

holds. By using (4.7) and (4.28), (4.29) it follows that

$$\begin{aligned} A_{k-1}(k:k+1,k-1) &= U_{k-1}(k:k+1,k-1) - p_{k-1}(k:k+1)q_{k-1}(k-1) \\ &= \begin{pmatrix} s_k \\ \rho_k \end{pmatrix} - \begin{pmatrix} c_k \\ p(k+1) \end{pmatrix} q^{(1)}(k-1) \end{aligned}$$

and, therefore, to get zero entry in the position $(k+1,k-1)$ in the matrix $A'_k = \tilde{Q}^*_k A_{k-1}$ one should take Q_k such that (4.13) holds. Next by using (4.13), (4.14) and (4.16), (4.17) we get

$$U'_k(1:k+1,k-1:N) = \begin{pmatrix} G^U_{k-1}h_U(k-1) & G^U_{k-1}\tilde{b}(k) & G^U_{k-1}\hat{b}(k)H_{k+1} \\ \sigma^{(1)}_{k-1} & \gamma_k & w_k H_{k+1} \\ c_{k+1}q^{(1)}(k-1) & f_{k+1} & v_{k+1}H_{k+1} \end{pmatrix}. \tag{4.44}$$

From (4.28), (4.29), (4.7) and the fact that A'_k is upper Hessenberg we deduce that $U'_k(k+2,k) = p(k+2)\theta_k$. Hence, by using (4.31), (2.12), (2.11) together with (4.44) we obtain

$$U'_k(1:k+2,k:N) = \begin{pmatrix} G^U_{k-1}\tilde{b}(k) & G^U_{k-1}\hat{b}(k)h(k+1) & G^U_{k-1}\hat{b}(k)b(k+1)H_{k+2} \\ \gamma_k & w_k h(k+1) & w_k b(k+1)H_{k+2} \\ f_{k+1} & v_{k+1}h(k+1) & v_{k+1}b(k+1)H_{k+2} \\ p(k+2)\theta_k & \sigma_{k+1} & g(k+2)H_{k+2} \end{pmatrix}.$$

From the decomposition

$$U'_k(1:k,k:N) = \begin{pmatrix} G^U_{k-1}\tilde{b}(k) & G^U_{k-1}\hat{b}(k) \\ \gamma_k & w_k \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & h(k+1) & b(k+1) \end{pmatrix} \begin{pmatrix} I_2 & 0 \\ 0 & H_{k+2} \end{pmatrix}$$

by applying (4.34) and (4.36) one finds that

$$U_k(1:k,k:N) = G^U_k \begin{pmatrix} h_U(k) & \tilde{b}(k+1) & \hat{b}(k+1)H_{k+2} \end{pmatrix}. \tag{4.45}$$

From (4.15) it is also seen that

$$U_k(k+1:k+2,k:N) = \begin{pmatrix} s_{k+1} & \eta_{k+1} & v_{k+1}b(k+1)H_{k+2} \\ \rho_{k+1} & \beta_{k+1} & g(k+2)H_{k+2} \end{pmatrix}. \tag{4.46}$$

By combining (4.45) and (4.46) together we obtain (4.38).

Now (4.38) with $k = N-2$ gives

$$U_{N-2}(1:N,N-2:N) = \begin{pmatrix} G^U_{N-2}h_U(N-2) & G^U_{N-2}\tilde{b}(N-1) & G^U_{N-2}\hat{b}(N-1)h(N) \\ s_{N-1} & \eta_{N-1} & v_{N-1}b(N-1)h(N) \\ \rho_{N-1} & \beta_{N-1} & g(N)h(N) \end{pmatrix}. \tag{4.47}$$

In the same way as above we obtain (4.18). By using (4.18), (4.19) and (4.21), (4.22) we deduce that

$$U'_{N-1}(1:N, N-2:N)$$
$$= \begin{pmatrix} G^U_{N-2}h_U(N-2) & G^U_{N-2}\tilde{b}(N-1) & G^U_{N-2}\hat{b}(N-1)h(N) \\ \sigma^{(1)}_{N-2} & \gamma_{N-1} & w_{N-1}h(N) \\ p^{(1)}(N)q^{(1)}(N-2) & f_N & v_N h(N) \end{pmatrix}. \tag{4.48}$$

From (4.34) with $k = N-1$ it follows that

$$U'_{N-1}(1:N-1, N-1:N) = G^U_{N-1}\begin{pmatrix} 1 & 0 \\ 0 & h(N) \end{pmatrix}.$$

By using (4.20) and (4.37) we obtain

$$U_{N-1}(1:N, N-1:N) = \begin{pmatrix} G^U_{N-1}h_U(N-1) & G^U_{N-1}b_U(N-1) \\ s_N & \eta_N \end{pmatrix}. \tag{4.49}$$

In the view of (4.34) with $k = N$ and $h_U(N) = 1$ this implies that

$$U_{N-1}(1:N, N) = G^U_N h_U(N). \tag{4.50}$$

Thus, by using (4.44), (4.48) and (4.30) we get

$$\sigma^{(1)}_{k-1} = U'(k, k-1) = U^{(1)}(k, k-1), \quad k = 2, \dots, N-1$$

and using (4.49) and (4.32) we have

$$\sigma^{(1)}_{N-1} = s_N = U_{N-1}(N, N-1) = U^{(1)}(N, N-1).$$

Hence it follows that $\sigma^{(1)}_k$ $(k = 1, \dots, N-1)$ are the subdiagonal entries of the matrix $U^{(1)}$. Finally, from (4.38), (4.49), (4.50) and (4.32) we deduce that

$$U^{(1)}(1:k, k) = G^U_k h_U(k), \quad k = 1, \dots, N.$$

By virtue of Lemma 2.2 this relation implies that $g_U(i), h_U(i)$ $(i = 1, \dots, N)$, and $b_U(k)$ $(k = 1, \dots, N-1)$ are upper triangular generators of the matrix $U^{(1)}$. The orders of these generators equal $r'_k = r_{k+1} + 1$, $k = 1, \dots, N-1$, $r'_N = 1$. By applying Theorem 3.1 we obtain upper triangular generators $g^{(1)}(i), h^{(1)}(i)$ $(i = 1, \dots, N)$, $b^{(1)}(k)$ $(k = 1, \dots, N-1)$ of the matrix $U^{(1)}$ with orders defined in (4.27) and this concludes the proof. □

The structured single-shift QR iteration applied to an input matrix $A = A_0 \in \mathcal{H}_N$ requires a linear number of arithmetic operations per step by using linear storage. Experimental comparisons with customary eigensolvers are shown in Section 6.

5. The structured QR iteration: the double-shift case

The double-shift technique is employed to compute a pair of complex conjugate eigenvalues of a real upper Hessenberg matrix by using real arithmetic only. A double-shift step of the QR iteration (2.2) consists of the computation of a new iterate $A^{(1)}$ generated from $A = A^{(0)}$ with shift polynomial $q(z) = (z-\alpha)(z-\bar{\alpha}) = z^2 - sz + t$, $s, t \in \mathbb{R}$. The implicit version proceeds by finding the unitary matrix Q of the form

$$Q = \tilde{Q}_1\tilde{Q}_2 \cdots \tilde{Q}_{N-2}\tilde{Q}_{N-1}, \tag{5.1}$$

where $\tilde{Q}_i \in \mathbb{C}^{N\times N}$ are unitary matrices satisfying

$$\tilde{Q}_i = I_{i-1} \oplus Q_i \oplus I_{N-i-2}, \quad i = 1, \ldots, N-2, \qquad \tilde{Q}_{N-1} = I_{N-2} \oplus Q_{N-1}. \tag{5.2}$$

These matrices are determined so that $(\tilde{Q}_1^* q(A))(2:3,1) = (0\ 0)^T$ and $A^{(1)} = Q^* A Q$ is an upper Hessenberg matrix.

By using a standard bulge-chasing approach one computes the unitary matrices Q_i ($i = 1, \ldots, N-1$) and then the matrix $A^{(1)}$ as follows. At the first step we determine the 3×3 unitary matrix Q_1 from the condition

$$Q_1^*(q(A)(1:3,1)) = \begin{pmatrix} * & 0 & 0 \end{pmatrix}^T. \tag{5.3}$$

Next we compute the matrix $A_1 = \tilde{Q}_1^* A \tilde{Q}_1$ which is similar to the matrix A and contains nonzero entries in the positions (3, 1) and (4, 1). We choose the matrix Q_2 in order to annihilate these entries, i.e., to obtain the matrix $A_2' = \tilde{Q}_2^* A_1$ with zero entries below the first subdiagonal in the first column. Then we compute the matrix $A_2 = A_2'\tilde{Q}_2$. The matrix A_2 is similar to the matrix A and contains nonzero entries in the positions (4, 2), (5, 2). We choose the matrix Q_3 in order to annihilate these entries, i.e., to obtain the matrix $A_3' = \tilde{Q}_3^* A_2$ with zero entries below the first subdiagonal in the second column. Then we compute the matrix $A_3 = A_3'\tilde{Q}_3$. The matrix A_3 is similar to the matrix A and contains nonzero entries in the positions (5, 3), (6, 3). We continue this procedure for $k = 4, 5, \ldots, N-2$ and obtain the matrix A_{N-2} with a nonzero entry in the $(N, N-2)$ position. To annihilate this entry we determine a Givens rotation matrix Q_{N-1} and obtain $A_{N-1}' = \tilde{Q}_{N-1}^* A_{N-2}$ and finally the upper Hessenberg matrix $A^{(1)} = A_{N-1} = A_{N-1}' Q_{N-1}$. Thus we obtain the sequence (4.3) to determine the new iterate $A^{(1)}$ in the double-shift case.

The next result reformulates the process in terms of generators by providing a structured QR iteration for the double-shift step applied to an input matrix $A \in \mathcal{H}_N$.

Theorem 5.1. *Let $A = U - pq^T$ be a real matrix from the class $\mathcal{H}_N$ with vectors of perturbation $p = \mathrm{col}(p(i))_{i=1}^N$, $q = \mathrm{col}(q(i))_{i=1}^N$, upper triangular generators $g(i), h(i)$ ($i = 1, \ldots, N$), $b(k)$ ($k = 1, \ldots, N-1$) of orders r_k ($k = 1, \ldots, N$) and subdiagonal entries σ_k ($k = 1, \ldots, N-1$) of the orthogonal matrix U, and let s, t be real numbers. Then the unitary matrices Q_k, $k = 1, \ldots, N-1$, the vectors of perturbation $p^{(1)} = \mathrm{col}(p^{(1)}(i))_{i=1}^N$, $q^{(1)} = \mathrm{col}(q^{(1)}(i))_{i=1}^N$ of the upper Hessenberg matrix $A^{(1)} = Q^*AQ$, as well as subdiagonal entries $\sigma_k^{(1)}$ ($k = 1, \ldots, N-1$) and the upper triangular generators $g^{(1)}(i), h^{(1)}(i)$ ($i = 1, \ldots, N$), $b^{(1)}(k)$ ($k = 1, \ldots, N-1$) with orders not greater than two of the unitary matrix $U^{(1)}$ from (4.8) are determined by the following procedure.*

1. *Compute the unitary matrices Q_k, $k = 1, \ldots, N-1$, the vectors of perturbation $p^{(1)} = \mathrm{col}(p^{(1)}(i))_{i=1}^N$, $q^{(1)} = \mathrm{col}(q^{(1)}(i))_{i=1}^N$ of the upper Hessenberg matrix $A^{(1)} = Q^*AQ$ and the subdiagonal entries $\sigma_k^{(1)}$ ($k = 1, \ldots, N-1$) of the unitary matrix $U^{(1)}$, by performing these steps.*
   (a) *Set*

$$\begin{aligned} &a_{11} = g(1)h(1) - p(1)q(1), \qquad a_{12} = g(1)b(1)h(2) - p(1)q(2),\\ &a_{22} = g(2)h(2) - p(2)q(2),\\ &a_{21} = \sigma_1 - p(2)q(1), \qquad a_{32} = \sigma_2 - p(3)q(2), \end{aligned} \tag{5.4}$$

compute

$$v = \begin{pmatrix} a_{11}^2 + a_{12}a_{21} - sa_{21} + t & a_{21}(a_{11} + a_{22} - s) & a_{32}a_{21} \end{pmatrix}^T,$$

and determine the 3×3 *orthogonal matrix* Q_1 *from the condition*

$$Q_1^* v = \begin{pmatrix} * & 0 & 0 \end{pmatrix}^T. \tag{5.5}$$

Set

$$\theta_2 = \begin{pmatrix} q(1) & q(2) \end{pmatrix}. \tag{5.6}$$

Compute

$$\begin{pmatrix} \gamma_1 & w_1 \\ f_3 & v_3 \end{pmatrix} = Q_1^* \begin{pmatrix} g(1)h(1) & g(1)b(1)h(2) & g(1)b(1)b(2) \\ \sigma_1 & g(2)h(2) & g(2)b(2) \\ p(3)q(1) & \sigma_2 & g(3) \end{pmatrix}, \tag{5.7}$$

$$\begin{pmatrix} s_2 & \eta_2 \\ \rho_2 & \beta_2 \end{pmatrix} = \begin{pmatrix} f_3 & v_3 h(3) \\ p(4)\theta_2 & \sigma_3 \end{pmatrix} Q_1, \tag{5.8}$$

$$\begin{pmatrix} p^{(1)}(1) \\ c_3 \end{pmatrix} = Q_1^* \begin{pmatrix} c_2 \\ p(3) \end{pmatrix}, \tag{5.9}$$

$$H_1 = \begin{pmatrix} f_3 - c_3\theta_2 & v_3 h(3) - c_3 q(3) \\ 0_{1\times 2} & \sigma_3 - p(4)q(3) \end{pmatrix} Q_1, \tag{5.10}$$

$$\begin{pmatrix} q^{(1)}(1) & \theta_3 \end{pmatrix} = \begin{pmatrix} q(1) & q(2) & q(3) \end{pmatrix} Q_1 \tag{5.11}$$

with matrices $\gamma_1, w_1, f_3, v_3, s_2, \eta_2, \rho_2, \beta_2, c_3, \theta_3, H_1$ *of sizes* $1 \times 2, 1 \times r_3, 2 \times 2, 2 \times r_3, 2 \times 1, 2 \times 2, 1 \times 1, 1 \times 2, 2 \times 1, 1 \times 2, 3 \times 3$ *respectively.*

(b) *For* $k = 2, \ldots, N-3$ *perform the following computation. Determine the* 3×3 *unitary matrix* Q_k *and the number* δ_{k-1} *such that*

$$Q_k^* H_{k-1}(1:3, 1) = \begin{pmatrix} \delta_{k-1} \\ 0 \\ 0 \end{pmatrix}. \tag{5.12}$$

Compute

$$\begin{pmatrix} \gamma_k & w_k \\ f_{k+2} & v_{k+2} \end{pmatrix} = Q_k^* \begin{pmatrix} \eta_k & v_{k+1} b(k+1) \\ \beta_k & g(k+2) \end{pmatrix}, \tag{5.13}$$

$$\begin{pmatrix} s_{k+1} & \eta_{k+1} \\ \rho_{k+1} & \beta_{k+1} \end{pmatrix} = \begin{pmatrix} f_{k+2} & v_{k+2} h(k+2) \\ p(k+3)\theta_{k+1} & \sigma_{k+2} \end{pmatrix} Q_k, \tag{5.14}$$

$$\begin{pmatrix} p^{(1)}(k) \\ c_{k+2} \end{pmatrix} = Q_k^* \begin{pmatrix} c_{k+1} \\ p(k+2) \end{pmatrix}, \tag{5.15}$$

$$H_k = \begin{pmatrix} f_{k+2} - c_{k+2}\theta_{k+1} & v_{k+2} h(k+2) - c_{k+2} q(k+2) \\ 0_{1\times 2} & \sigma_{k+2} - p(k+3)q(k+2) \end{pmatrix} Q_k, \tag{5.16}$$

$$\begin{pmatrix} q^{(1)}(k) & \theta_{k+2} \end{pmatrix} = \begin{pmatrix} \theta_{k+1} & q(k+2) \end{pmatrix} Q_k, \tag{5.17}$$

$$\sigma_{k-1}^{(1)} = \delta_{k-1} + p^{(1)}(k) q^{(1)}(k-1) \tag{5.18}$$

with matrices $\gamma_k, w_k, f_{k+2}, v_{k+2}, s_{k+1}, \eta_{k+1}, \rho_{k+1}, \beta_{k+1}, c_{k+2}, \theta_{k+2}, H_k$ *of sizes* $1 \times 2, 1 \times r_{k+2}, 2 \times 2, 2 \times r_{k+2}, 2 \times 1, 2 \times 2, 1 \times 1, 1 \times 2, 2 \times 1, 1 \times 2, 3 \times 3$ *respectively.*

(c) *Find the* 3×3 *unitary matrix* Q_{N-2} *and the number* δ_{N-2} *such that*

$$Q_{N-2}^* H_{N-3}(1:3, 1) = \begin{pmatrix} \delta_{N-3} \\ 0 \\ 0 \end{pmatrix}. \tag{5.19}$$

Compute

$$\begin{pmatrix} \gamma_{N-2} & w_{N-2} \\ f_N & v_N \end{pmatrix} = Q_{N-2}^* \begin{pmatrix} \eta_{N-2} & v_{N-1}b(N-1) \\ \beta_{N-2} & g(N) \end{pmatrix}, \tag{5.20}$$

$$\begin{pmatrix} s_{N-1} & \eta_{N-1} \end{pmatrix} = \begin{pmatrix} f_N & v_N h(N) \end{pmatrix} Q_{N-2}, \tag{5.21}$$

$$\begin{pmatrix} p^{(1)}(N-2) \\ c_N \end{pmatrix} = Q_{N-2}^* \begin{pmatrix} c_{N-1} \\ p(N) \end{pmatrix}, \tag{5.22}$$

$$H_{N-2} = \begin{pmatrix} f_N - c_N\theta_{N-1} & v_N h(N) - c_N q(N) \end{pmatrix} Q_{N-2}, \tag{5.23}$$

$$\begin{pmatrix} q^{(1)}(N-2) & \theta_N \end{pmatrix} = \begin{pmatrix} \theta_{N-1} & q(N) \end{pmatrix} Q_{N-2}, \tag{5.24}$$

$$\sigma_{N-3}^{(1)} = \delta_{N-3} + p^{(1)}(N-2)q^{(1)}(N-3) \tag{5.25}$$

with matrices $\gamma_{N-2}, w_{N-2}, f_N, v_N, s_{N-1}, \eta_{N-1}, \rho_{N-1}, \beta_{N-1}, c_N, \theta_N$, and H_{N-2} of sizes $1\times 2, 1\times r_N, 2\times 2, 2\times r_N, 2\times 1, 2\times 2, 1\times 1, 1\times 2, 2\times 1, 1\times 2, 2\times 2$ respectively.

(d) *Compute the Givens rotation matrix Q_{N-1} and the number δ_{N-2} such that*

$$Q_{N-1}^* H_{N-2}(1:2,1) = \begin{pmatrix} \delta_{N-2} \\ 0 \end{pmatrix}. \tag{5.26}$$

Compute

$$\begin{pmatrix} \gamma_{N-1} \\ f_{N+1} \end{pmatrix} = Q_{N-1}^* \eta_{N-1}, \tag{5.27}$$

$$\begin{pmatrix} s_N & \eta_N \end{pmatrix} = f_{N+1} Q_{N-1}, \tag{5.28}$$

$$\begin{pmatrix} p^{(1)}(N-1) \\ p^{(1)}(N) \end{pmatrix} = Q_{N-1}^* c_N, \qquad \begin{pmatrix} q^{(1)}(N-1) & q^{(1)}(N) \end{pmatrix} = \theta_N Q_{N-1}, \tag{5.29}$$

$$\sigma_{N-2}^{(1)} = \delta_{N-2} + p^{(1)}(N-1)q^{(1)}(N-2) \tag{5.30}$$

with two-dimensional rows γ_{N-1}, f_{N+1} and numbers s_N, η_N. Set $\sigma_{N-1}^{(1)} = s_N$.

2. *Determine the matrices $h_Q(k), b_Q(k), d_Q(k), g_Q(k)$ of sizes $2\times 1, 2\times 2, 1\times 1, 1\times 2$ from the partitions*

$$Q_k = \begin{pmatrix} h_Q(k) & b_Q(k) \\ d_Q(k) & g_Q(k) \end{pmatrix}, \quad k = 1,\dots,N-2 \tag{5.31}$$

and the two-dimensional columns $h_Q(N-1), b_Q(N-1)$ from the partition

$$Q_{N-1} = \begin{pmatrix} h_Q(N-1) & b_Q(N-1) \end{pmatrix}. \tag{5.32}$$

Set

$$\begin{aligned} &g_U(k) = \begin{pmatrix} \gamma_k & w_k \end{pmatrix}, \quad k = 1,\dots,N-2, \\ &g_U(N-1) = \gamma_{N-1}, \qquad g_U(N) = \eta_N, \end{aligned} \tag{5.33}$$

$$\begin{aligned} &h_U(k) = \begin{pmatrix} h_Q(k) \\ h(k+2)d_Q(k) \end{pmatrix}, \quad k = 1,\dots,N-2, \\ &h_U(N-1) = h_Q(N-1), \qquad h_U(N) = 1, \end{aligned} \tag{5.34}$$

$$\begin{aligned} &b_U(k) = \begin{pmatrix} b_Q(k) & 0 \\ h(k+2)g_Q(k) & b(k+2) \end{pmatrix}, \quad k = 1,\dots,N-3, \\ &b_U(N-2) = \begin{pmatrix} b_Q(N-2) \\ h(N)g_Q(N-2) \end{pmatrix}, \qquad b_U(N-1) = b_Q(N-1). \end{aligned} \tag{5.35}$$

3. *By using the algorithm from Theorem 3.1 compute upper triangular generators $g^{(1)}(i), h^{(1)}(i)$ $(i = 1, \dots, N)$, $b^{(1)}(k)$ $(k = 1, \dots, N-1)$ of orders*

$$t_k = 2, \quad k = 1, \dots, N-1, \qquad t_N = 1$$

of the matrix $U^{(1)}$.

Remark 5.2. The "bulge vector" $H_{k-1}(1:3,1)$, which is used in (5.12) to determine Q_k, can also, in principle, be computed as

$$H_{k-1}(1:3,1) = \begin{pmatrix} s_k - c_{k+1} q^{(1)}(k-1) \\ \rho_k - p(k+2) q^{(1)}(k-1) \end{pmatrix},$$

thus making the computation of H_k in (5.16) unnecessary. However, we prefer to compute the bulge vector as in (5.16) for reasons of numerical stability.

Proof. From the formulas (5.1), (5.2) it follows that the vectors p_k, q_k from (4.5), (4.6) have the form

$$p_k = (p^{(1)}(1), \dots, p^{(1)}(k), c_{k+2}, p(k+3), \dots, p(N))^T, \quad k = 1, \dots, N-2, \tag{5.36}$$

$$q_k = (q^{(1)}(1), \dots, q^{(1)}(k), \theta_{k+2}, q(k+3), \dots, q(N))^T, \quad k = 1, \dots, N-2, \tag{5.37}$$

and moreover the formulas (5.9), (5.11), (5.15), (5.17), (5.22), (5.24) and (5.29) yield the coordinates of the vectors of perturbation $p^{(1)} = Q^* p$, $q^{(1)} = Q^T q$. Furthermore from (4.4) one deduces the relations (4.30), (4.32) together with the equalities

$$U'_k(k+3:N, k+2:N) = U(k+3:N, k+2:N), \quad k = 1, \dots, N-3. \tag{5.38}$$

By using the matrices $g_U(k), b_U(k)$ define the matrices G_k^U via the recursive relations (4.33). From (5.33), (5.35) we have

$$\begin{gathered} G_1^U = \begin{pmatrix} \gamma_1 & w_1 \end{pmatrix}, \qquad G_k^U = \begin{pmatrix} G_{k-1}^U \tilde{b}(k) & G_{k-1}^U \hat{b}(k) \\ \gamma_k & w_k \end{pmatrix}, \quad k = 2, \dots, N-2, \\ G_{N-1}^U = \begin{pmatrix} G_{N-2}^U b_U(N-2) \\ \gamma_{N-1} \end{pmatrix}, \qquad G_N^U = \begin{pmatrix} G_{N-1}^U b_U(N-1) \\ \eta_N \end{pmatrix} \end{gathered} \tag{5.39}$$

with

$$\tilde{b}(k) = \begin{pmatrix} b_Q(k-1) \\ h(k+1) g_Q(k-1) \end{pmatrix}, \qquad \hat{b}(k) = \begin{pmatrix} 0_{1 \times r_{k+2}} \\ b(k+1) \end{pmatrix}, \quad k = 2, \dots, N-2. \tag{5.40}$$

Notice that from (5.31) one can rewrite the formulas (5.34), (5.35) in the form

$$\begin{gathered} \begin{pmatrix} h_U(k) & \tilde{b}(k+1) & \hat{b}(k+1) \end{pmatrix} = \begin{pmatrix} I_2 & 0 & 0 \\ 0 & h(k+2) & b(k+2) \end{pmatrix} \begin{pmatrix} Q_k & 0 \\ 0 & I_{r_{k+3}} \end{pmatrix}, \\ k = 1, \dots, N-3, \end{gathered} \tag{5.41}$$

$$\begin{pmatrix} h_U(N-2) & b_U(N-2) \end{pmatrix} = \begin{pmatrix} I_2 & 0 \\ 0 & h(N) \end{pmatrix} Q_{N-2}. \tag{5.42}$$

We prove by induction that the relations

$$A_k(k+1:k+3, k:k+2) = H_k, \quad k = 1, \dots, N-3 \tag{5.43}$$

with the matrices H_k $(k = 1, \ldots, N-3)$ defined in (5.10), (5.16) hold, the 3×3 unitary matrices $Q_k,\ k = 1, \ldots, N-3$ are determined via (5.5), (5.12) and, moreover, that the relations

$$U_k(1:k+3, k:N) = \begin{pmatrix} G_k^U h_U(k) & G_k^U \tilde{b}(k+1) & G_k^U \hat{b}(k+1) H_{k+3} \\ s_{k+1} & \eta_{k+1} & v_{k+2} b(k+2) H_{k+3} \\ \rho_{k+1} & \beta_{k+1} & g(k+3) H_{k+3} \end{pmatrix}, \quad k = 1, \ldots, N-3 \tag{5.44}$$

are fulfilled.

By using (2.6) and (2.9) we obtain the formulas (5.4). Hence to satisfy the condition (5.3) one should take the 3×3 unitary matrix Q_1 such that the condition (5.5) holds. From (2.6), (2.12) and (2.11) we find that $U(1:4, 1:N)$ can be specified as follows:

$$\begin{pmatrix} g(1)h(1) & g(1)b(1)h(2) & g(1)b(1)b(2)h(3) & g(1)b(1)b(2)b(3)H_4 \\ \sigma_1 & g(2)h(2) & g(2)b(2)h(3) & g(2)b(2)b(3)H_4 \\ p(3)q(1) & \sigma_2 & g(3)h(3) & g(3)b(3)H_4 \\ p(4)q(1) & p(4)q(2) & \sigma_3 & g(4)H_4 \end{pmatrix}.$$

From (5.7) and (5.6) we obtain

$$U_1'(1:4, 1:N) = \begin{pmatrix} \gamma_1 & w_1 h(3) & w_1 b(3) H_4 \\ f_3 & v_3 h(3) & v_3 b(3) H_4 \\ p(4)\theta_2 & \sigma_3 & g(4) H_4 \end{pmatrix}. \tag{5.45}$$

By using (4.3) and (4.7) we find that

$$A_1(2:4, 1:3) = A_1'(2:4, 1:3) Q_1 = [U_1'(2:4, 1:3) - p_1(2:4) q(1:3)] Q_1.$$

From (5.45), (5.36) and (5.6) this implies that

$$A_1(2:4, 1:3) = \left[\begin{pmatrix} f_3 & v_3 h(3) \\ p(4)\theta_2 & \sigma_3 \end{pmatrix} - \begin{pmatrix} c_3 \\ p(4) \end{pmatrix} \begin{pmatrix} \theta_2 & q(3) \end{pmatrix} \right] Q_1$$

which gives (5.43) with $k = 1$.

The first row of the matrix U_1' can be expressed in the form

$$U_1'(1, 1:N) = \begin{pmatrix} \gamma_1 & w_1 \end{pmatrix} \begin{pmatrix} I_2 & 0 & 0 \\ 0 & h(3) & b(3) \end{pmatrix} \begin{pmatrix} I_3 & 0 \\ 0 & H_4 \end{pmatrix}. \tag{5.46}$$

Using (5.46) and (5.41) with $k = 1$ yields

$$\begin{aligned} U_1(1, 1:N) &= G_1^U \begin{pmatrix} h_U(1) & \tilde{b}(2) & \hat{b}(2) \end{pmatrix} \begin{pmatrix} I_2 & 0 \\ 0 & H_4 \end{pmatrix} \\ &= \begin{pmatrix} G_1^U h_U(1) & G_1^U \tilde{b}(2) & G_1^U \hat{b}(2) H_4 \end{pmatrix}. \end{aligned} \tag{5.47}$$

From (5.8) we deduce that

$$U_1(2:4, 1:N) = \begin{pmatrix} s_2 & \eta_2 & v_3 b(3) H_4 \\ \rho_2 & \beta_2 & g(4) H_4 \end{pmatrix}. \tag{5.48}$$

By combining (5.47) and (5.48) together we obtain (5.44) with $k = 1$.

Let us now assume that for some k with $2 \le k \le N-4$ the relations

$$A_{k-1}(k:k+2, k-1:k+1) = H_{k-1} \tag{5.49}$$

and

$$U_{k-1}(1:k+1,k-1:N) = \begin{pmatrix} G^U_{k-1}h_U(k-1) & G^U_{k-1}\tilde{b}(k) & G^U_{k-1}\hat{b}(k)H_{k+2} \\ s_k & \eta_k & v_{k+1}b(k+1)H_{k+2} \\ \rho_k & \beta_k & g(k+2)H_{k+2} \end{pmatrix} \tag{5.50}$$

hold. Then from (5.49) it follows that in order to get zero entries in the positions $(k+1,k-1)$, $(k+2,k-1)$ in the matrix $A'_k = \tilde{Q}^*_k A_{k-1}$ one should take Q_k such that (5.12) holds. By using the equalities (5.49), (5.50), (4.7), (5.36) and (5.37) there follows that

$$\begin{aligned} \begin{pmatrix} s_k \\ \rho_k \end{pmatrix} &= A_{k-1}(k:k+2,k-1) + p_{k-1}(k:k+2)q_{k-1}(k-1) \\ &= H_{k-1}(:,1) + \begin{pmatrix} c_{k+1} \\ p(k+2) \end{pmatrix} q^{(1)}(k-1). \end{aligned}$$

Hence from (5.12), (5.15) and (5.18) we obtain

$$\begin{aligned} Q^*_{k-1}\begin{pmatrix} s_k \\ \rho_k \end{pmatrix} &= \begin{pmatrix} \delta_{k-1} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} p^{(1)}(k+1) \\ c_{k+2} \end{pmatrix} q^{(1)}(k-1) \\ &= \begin{pmatrix} \sigma^{(1)}_{k-1} \\ c_{k+2}q^{(1)}(k-1) \end{pmatrix}. \end{aligned} \tag{5.51}$$

Thus by virtue of (5.50), (5.51) and (5.14) we find that

$$U'_k(1:k+2,k-1:N) = \begin{pmatrix} G^U_{k-1}h_U(k-1) & G^U_{k-1}\tilde{b}(k) & G^U_{k-1}\hat{b}(k)H_{k+2} \\ \sigma^{(1)}_{k-1} & \gamma_k & w_k H_{k+2} \\ c_{k+2}q^{(1)}(k-1) & f_{k+2} & v_{k+2}H_{k+2} \end{pmatrix}. \tag{5.52}$$

By using (5.36), (5.37), (4.7) and the fact that A'_k has all zero entries below the first subdiagonal (except that one in the position $(k+2,k)$) we deduce that $U'_k(k+3,k) = p(k+3)\theta_{k+1}$. Hence by using (5.38), (2.12), (2.11) together with (5.52) we get

$$U'_k(1:k+3,k:N) = \begin{pmatrix} G^U_{k-1}\tilde{b}(k) & G^U_{k-1}\hat{b}(k)h(k+2) & G^U_{k-1}\hat{b}(k)b(k+2)H_{k+3} \\ \gamma_k & w_k h(k+2) & w_k b(k+2)H_{k+3} \\ f_{k+2} & v_{k+2}h(k+2) & v_{k+2}b(k+2)H_{k+3} \\ p(k+3)\theta_{k+1} & \sigma_{k+2} & g(k+3)H_{k+3} \end{pmatrix}. \tag{5.53}$$

By using (4.3) and (4.7) we find that

$$\begin{aligned} A_k(k+1:k+3,k:k+2) &= A'_k(k+1:k+3,k:k+2)Q_k \\ &= [U'_k(k+1:k+3,k:k+2) \\ &\quad - p_k(k+1:k+3)q_{k-1}(k:k+2)]Q_k. \end{aligned}$$

In the view of (5.53), (5.36) and (5.37) this gives

$$A_k(k+1:k+3,k:k+2) = \left[\begin{pmatrix} f_{k+2} & v_{k+2}h(k+2) \\ p(k+3)\theta_{k+1} & \sigma_{k+2} \end{pmatrix} - \begin{pmatrix} c_{k+2} \\ p(k+3) \end{pmatrix} \begin{pmatrix} \theta_{k+1} & q(k+2) \end{pmatrix}\right] Q_k$$

which implies (5.43).

Now let us consider the representation of $U'_k(1:k,k:N)$ as

$$\begin{pmatrix} G^U_{k-1}\tilde{b}(k) & G^U_{k-1}\hat{b}(k) \\ \gamma_k & w_k \end{pmatrix} \begin{pmatrix} I_2 & 0 & 0 \\ 0 & h(k+2) & b(k+2) \end{pmatrix} \begin{pmatrix} I_3 & 0 \\ 0 & H_{k+3} \end{pmatrix}.$$

By applying (5.39) and (5.41) we obtain

$$U_k(1:k,k:N) = G^U_k \begin{pmatrix} h_U(k) & \tilde{b}(k+1) & \hat{b}(k+1)H_{k+3} \end{pmatrix}. \tag{5.54}$$

By using (5.14) we get

$$U_k(k+1:k+3,k:N) = \begin{pmatrix} s_{k+1} & \eta_{k+1} & v_{k+2}b(k+2)H_{k+3} \\ \rho_{k+1} & \beta_{k+1} & g(k+3)H_{k+3} \end{pmatrix}. \tag{5.55}$$

Combining (5.54) and (5.55) together gives (5.44).

From (5.44) with $k = N-3$ we deduce that

$$U_{N-3}(1:N,N-3:N) = \begin{pmatrix} G^U_{N-3}h_U(N-3) & G^U_{N-3}\tilde{b}(N-2) & G^U_{N-3}\hat{b}(N-2)h(N) \\ s_{N-2} & \eta_{N-2} & v_{N-1}b(N-1)h(N) \\ \rho_{N-2} & \beta_{N-2} & g(N)h(N) \end{pmatrix}. \tag{5.56}$$

In the same way as above we obtain (5.19) and

$$U'_{N-2}(1:N,N-3:N) = \begin{pmatrix} G^U_{N-3}h_U(N-3) & G^U_{N-3}\tilde{b}(N-2) & G^U_{N-3}\hat{b}(N-2)h(N) \\ \sigma^{(1)}_{N-3} & \gamma_{N-2} & w_{N-2}h(N) \\ c_N q^{(1)}(N-2) & f_N & v_N h(N) \end{pmatrix}. \tag{5.57}$$

By using (4.3) and (4.7) we find that

$$\begin{aligned} A_{N-2}(N-1:N,N-2:N) &= A'_{N-2}(N-1:N,N-2:N)Q_{N-2} \\ &= [U'_{N-2}(N-1:N,N-2:N) \\ &\quad - p_{N-2}(N-1:N)q_{N-3}(N-2:N)]Q_{N-2}. \end{aligned}$$

From this relation in the view of (5.57), (5.36) and (5.37) we obtain

$$A_{N-2}(N-1:N,N-2:N) = \left[\begin{pmatrix} f_N & v_N h(N) \end{pmatrix} - c_N \begin{pmatrix} \theta_{N-1} & q(N) \end{pmatrix}\right] Q_{N-2}$$

which implies

$$A_{N-2}(N-1:N,N-2:N) = H_{N-2} \tag{5.58}$$

with the matrix H_{N-2} defined in (5.23). Next from (5.57) and (5.39) with $k = N-2$ we have

$$U'_{N-2}(1:N-2,N-2:N) = G^U_{N-2} \begin{pmatrix} I_2 & 0 \\ 0 & h(N) \end{pmatrix}.$$

By using (5.21) and (5.42) we get

$$U_{N-2}(1:N-2,N-2:N) = \begin{pmatrix} G^U_{N-2}h_U(N-2) & G^U_{N-2}b_U(N-2) \\ s_{N-1} & \eta_{N-1} \end{pmatrix}. \tag{5.59}$$

From (5.58) it follows that in order to get zero entry in the position $(N, N-2)$ in the matrix $A'_{N-1} = \tilde{Q}^*_k A_{N-2}$ one should take Q_{N-1} such that (5.26) holds. By using the equalities (5.58), (5.59), (4.7) and (5.36), (5.37) we find that

$$\begin{aligned} s_{N-1} &= A_{N-2}(N-1:N, N-2) + p_{N-2}(N-1:N)q_{N-2}(N-2) \\ &= H_{N-2}(:,1) + c_N q^{(1)}(N-2). \end{aligned}$$

Hence from (5.26), (5.29) and (5.30) we obtain

$$\begin{aligned} Q^*_{N-1}s_{N-1} &= \begin{pmatrix} \delta_{N-1} \\ 0 \end{pmatrix} + \begin{pmatrix} p^{(1)}(N-1) \\ p^{(1)}(N) \end{pmatrix} q^{(1)}(N-2) \\ &= \begin{pmatrix} \sigma^{(1)}_{N-2} \\ p^{(1)}(N)q^{(1)}(N-2) \end{pmatrix}. \end{aligned} \tag{5.60}$$

Thus by using (5.59), (5.60) and (5.27) we get

$$U'_{N-1}(1:N,N-2:N) = \begin{pmatrix} G^U_{N-2}h_U(N-2) & G^U_{N-2}b_U(N-2) \\ \sigma^{(1)}_{N-2} & \gamma_{N-1} \\ p^{(1)}(N)q^{(1)}(N-2) & f_{N+1} \end{pmatrix}. \tag{5.61}$$

From (5.39) with $k = N-1$ we have

$$U'_{N-1}(1:N,N-1:N) = \begin{pmatrix} G^U_{N-1}Q_{N-1} \\ f_{N+1}Q_{N-1} \end{pmatrix} \cdot \begin{pmatrix} I_2 & 0 \\ 0 & h(N) \end{pmatrix}.$$

Hence, (5.28), (5.32) and (5.34), (5.35) with $k = N-1$ imply

$$U_{N-1}(1:N,N-1:N) = \begin{pmatrix} G^U_{N-1}h_U(N-1) & G^U_{N-1}b_U(N-1) \\ s_N & \eta_N \end{pmatrix}. \tag{5.62}$$

In the view of (5.39) with $k = N$ and $h_U(N) = 1$ this relation gives

$$U_{N-1}(1:N,N) = G^U_N h_U(N). \tag{5.63}$$

Thus, by virtue of (5.52), (5.57), (5.61) and (4.30) we get

$$\sigma^{(1)}_{k-1} = U'(k,k-1) = U^{(1)}(k,k-1), \quad k = 2,\ldots,N-1$$

and from (5.62) and (4.32) this yields

$$\sigma^{(1)}_{N-1} = s_N = U_{N-1}(N,N-1) = U^{(1)}(N,N-1).$$

Hence it follows that $\sigma^{(1)}_k$ $(k = 1,\ldots,N-1)$ are the subdiagonal entries of the matrix $U^{(1)}$.

Furthermore, by using (5.44), (5.59), (5.63) and (4.32) we have

$$U^{(1)}(1:k,k) = G^U_k h_U(k), \quad k = 1,\ldots,N.$$

From Lemma 2.2 this means that $g_U(i), h_U(i)$ $(i = 1,\ldots,N)$, $b_U(k)$ $(k = 1,\ldots,N-1)$ are upper triangular generators of the matrix $U^{(1)}$. The orders of these generators equal $r'_k =$

$r_{k+2}+1,\ k=1,\ldots,N-2,\ r'_{N-1}=2,\ r'_N=1$. By applying Theorem 3.1 we obtain upper triangular generators $g^{(1)}(i), h^{(1)}(i)$ $(i=1,\ldots,N)$, $b^{(1)}(k)$ $(k=1,\ldots,N-1)$ of the matrix $U^{(1)}$ with orders defined in (4.27). □

Analogously with the single-shift case we may conclude that the structured double-shift QR iteration applied to an input matrix $A = A_0 \in \mathcal{H}_N$ requires a linear number of arithmetic operations per step by using linear storage. Experimental comparisons with customary eigensolvers are also shown in Section 6.

6. Numerical tests

In order to test the performance of the proposed structured QR iterations, we implemented the single-shift strategy in MATLAB and in Fortran 90/95, and the double-shift strategy in MATLAB. The Fortran implementation has been used for timing comparisons. The codes can be downloaded from the URL

http://www.unilim.fr/pages_perso/paola.boito/software.html

A crucial point in the implementation of the QR method is deflation. Deflation occurs when the current iterate matrix A is numerically reducible, that is, when a subdiagonal entry happens to be negligible. In this case the eigenvalue problem splits into two subproblems of smaller size. If the negligible subdiagonal entry is in last or last-but-one position, then one eigenvalue or a pair of complex conjugate eigenvalues, respectively, has converged. In the present implementations:

- Deflation is performed according to the classical Wilkinson criterion, i.e., a subdiagonal entry $A(k+1,k)$ is considered small enough for deflation if $|A(k+1,k)| < \epsilon\,(|A(k,k)| + |A(k+1,k+1)|)$, where ϵ is the machine epsilon;
- Deflation is also performed when the product of two consecutive subdiagonal entries is small enough to ensure that the current iterate is numerically reducible; see e.g. [9,6,1] for details.

The accuracy and stability of our algorithms has been measured in the numerical experiments by computing backward and forward errors. Let $\lambda(z) = \sum_{j=0}^{N} c_j z^j = c_n \prod_{j=1}^{N}(z-\alpha_j)$ be a test polynomial; denote as $\{\tilde{\alpha}_j\}_{j=1,\ldots,N}$ the roots of $\lambda(z)$ computed by the method that is being analyzed, and as $\{\tilde{c}_j\}_{j=0,\ldots,N}$ the coefficients of a polynomial having $\{\tilde{\alpha}_j\}_{j=1,\ldots,N}$ as "exact" roots. Here the coefficients $\{\tilde{c}_j\}_{j=0,\ldots,N}$ are computed in MATLAB by using the high precision arithmetic environment.

We use the following definitions for errors:

- Absolute forward error: MaxEigAbs $= \max_{j=1,\ldots,N} \min_{k=1,\ldots,N} |\alpha_j - \tilde{\alpha}_k|$.
- Relative forward error: MaxEigRel $= \max_{\alpha_j\neq 0,\, j=1,\ldots,N} \min_{k=1,\ldots,N} \frac{|\alpha_j-\tilde{\alpha}_k|}{|\alpha_j|}$.
- Absolute backward error: MaxCoeffAbs $= \max_{j=0,\ldots,N} |c_j - \tilde{c}_j|$.
- Relative backward error: MaxCoeffRel $= \max_{c_j\neq 0,\, j=0,\ldots,N} \frac{|c_j-\tilde{c}_j|}{|c_j|}$.

For practical purposes, when $\lambda(z)$ is defined by its coefficients we take as "exact" roots $\{\alpha_j\}_{j=1,\ldots,n}$ the roots computed by LAPACK routines `ZGEEV` (for complex coefficients) or `DGEEV` (for real coefficients), or by the MATLAB command `roots`. On the other hand, if $\lambda(z)$ is defined by its roots, then we compute its "exact" coefficients $\{c_j\}_{j=0,\ldots,N}$ by using high precision arithmetic.

In Examples 1 and 2, we consider polynomials of large degree (random and cyclotomic), useful to check the growth of running time and the stability of the algorithm. The Fortran implementation of the single shift strategy is employed here, with the exception of Example 1bis.

Table 1
Average results for complex polynomials with random coefficients. Times are measured in seconds. "Old" refers to the algorithm in [1], "new" to the algorithm described in the present paper.

N	Time (old)	Error (old)	Time (new)	Error (new)	time_L
50	0.0230	8.50e−15	0.0196	6.35e−15	0.00950
100	0.0685	3.23e−14	0.0584	1.25e−14	0.0446
150	0.136	7.00e−14	0.109	1.63e−14	0.134
200	0.278	1.03e−13	0.223	2.94e−14	0.320
300	0.605	2.77e−13	0.505	3.95e−14	1.10
400	1.00	4.34e−13	0.754	4.69e−14	2.67
500	1.59	6.31e−13	1.17	8.66e−14	5.17
600	2.48	8.82e−13	1.92	1.02e−14	8.64
700	3.64	1.54e−12	2.64	1.07e−13	14.65
800	4.45	2.11e−12	3.01	1.21e−13	23.13
900	5.67	1.95e−12	4.30	1.36e−13	31.15
1000	6.99	2.27e−12	4.96	1.84e−13	40.44
3000	63.88	1.55e−11	50.00	6.67e−13	1250.32
5000	197.81	4.74e−11	154.66	1.81e−12	7595.66

The machine we used to run the Fortran code is a laptop with an AMD Turion processor and 2 GB RAM, equipped with the f95 compiler under Linux Ubuntu. We have observed, however, that the same code may give slightly different results on different machines. We have also compared the performance of the algorithm described in the present paper to the fast QR algorithm presented in [1] referred to as the "Old" algorithm. For the sake of comparison the time time_L reported by the LAPACK routine is also indicated.

Example 1. In this example we consider polynomials with complex coefficients whose real and imaginary parts are randomly chosen in the range $[-1, 1]$. Table 1 shows, for several values of the degree, the average errors and timings over 10 polynomials. The cases $N = 3000$ and $N = 5000$ are exceptions in that a single polynomial has been used. Further, we have computed a linear fit on logarithmic timings for our algorithm, for random polynomials of degrees $100, 200, 300, \dots, 2000$. The resulting slope is 2.01, which supports the theoretical result of $\mathcal{O}(N^2)$ overall complexity for approximating all the roots of a polynomial of degree N.

Example 1bis. We also test the Matlab implementation of the double shift strategy for random polynomials (this time with real coefficients): see Table 2. For each value of the degree, results are computed as an average over 10 polynomials.

Example 2. Given a degree N, consider the complex polynomial $\lambda_N(z) = z^N - i$, where i is the imaginary unit. Table 3 shows the errors and timings for several values of the degree N.

Example 3. This example presents ill-conditioned polynomials of small degree, which provide a test for backward stability and for the accuracy of computed results. We use here the Matlab implementation, in the double shift version except for the last polynomial, which has complex coefficients.

Following [7], we also explore the effect of balancing. Balancing amounts to replacing the original companion matrix A with DAD^{-1}, where D is a diagonal matrix. Ideally, the variation

Table 2
Average results for the Matlab double shift implementation applied to real polynomials with random coefficients; N_it denotes the average number of iterations per eigenvalue.

N	MaxEigAbs	N_it
100	1.03e−14	1.69
200	2.15e−14	1.58
300	4.35e−14	1.58
400	4.63e−14	1.50
500	9.43e−14	1.49
600	7.21e−14	1.53
700	1.05e−13	1.46

Table 3
Results for polynomials $\lambda_N(z) = z^N - i$. Times are measured in seconds. "Old" refers to the algorithm in [1], "new" to the algorithm described in the present paper.

N	Time (old)	Error (old)	Time (new)	Error (new)	time_L
50	0.0450	6.82e−15	0.0360	4.72e−15	0.0340
100	0.117	7.97e−15	0.0680	1.25e−15	0.170
150	0.273	1.05e−14	0.143	1.40e−14	0.528
200	0.367	1.99e−14	0.218	1.66e−14	1.35
250	0.497	4.46e−14	0.320	1.77e−14	2.16
300	0.789	7.08e−14	0.557	2.22e−14	3.21
400	1.27	1.46e−13	0.815	2.66e−14	9.53
500	1.96	2.39e−13	1.46	4.94e−14	23.57
600	3.06	4.10e−13	2.18	2.40e−14	27.48
700	4.49	4.26e−13	2.90	3.58e−14	69.16
800	5.53	7.10e−13	3.99	4.13e−14	64.67
900	7.57	7.51e−13	4.94	1.57e−13	83.31
1000	8.84	9.02e−13	6.00	8.92e−14	168.04

in magnitude of the elements of the new matrix should be smaller than for A, thus improving the performance of a numerical rootfinder. For a structured approach, the matrix D needs to be chosen of the form $D = \mathrm{diag}(\beta, \beta^2, \beta^N)$ for a suitable parameter β which amounts to a scaling of the polynomial, i.e., to compute the roots of the scaled polynomial $\hat{\lambda}(z) = \lambda(\beta z)$. We therefore obtain

$$DAD^{-1} = \beta \begin{pmatrix} 0 & 0 & \cdots & 0 & -\dfrac{\lambda_0}{\beta^N} \\ 1 & 0 & \cdots & 0 & -\dfrac{\lambda_1}{\beta^{N-1}} \\ 0 & 0 & \cdots & 0 & -\dfrac{\lambda_2}{\beta^{N-2}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -\dfrac{\lambda_{N-1}}{\beta} \end{pmatrix} = \beta \hat{A},$$

where $\hat{A}$ is the companion matrix associated with $\hat{\lambda}(z)$. Hence, the eigenvalues of A can be easily recovered from the eigenvalues of $\hat{A}$ via multiplication by β. We will not go into details as to

Table 4
Results for polynomial 1 (Wilkinson polynomial).

	$\beta = 1$	$\beta = 2^3$	$\beta = 2^{-7}$	eig	eig nobal
MaxEigAbs	4.33e−3	8.25e−2	9.90e−3	6.53e−2	6.78e−2
MaxEigRel	3.33e−4	5.89e−3	7.63e−4	4.66e−3	4.52e−3
MaxCoeffAbs	9.96e+5	8.75e+5	4.40e+6	4.75e+4	7.21e+4
MaxCoeffRel	8.44e−14	1.55e−13	5.26e−13	3.99e−15	8.88e−15
MaxCondEig	3.23e+27	2.22e+15	3.95e+66		

Table 5
Results for polynomial 2.

	$\beta = 1$	$\beta = 2^{-7}$	eig	eig nobal
MaxEigAbs	5.09e−12	3.01e−12	2.05e−12	4.30e−13
MaxEigRel	3.27e−12	2.01e−12	1.36e−12	6.50e−13
MaxCoeffAbs	6.70e−12	1.60e−10	2.14e−12	1.19e−11
MaxCoeffRel	7.81e−13	1.19e−11	6.65e−14	9.88e−13
MaxCondEig	2.63e+5	1.95e+40		

the most effective criteria for choosing β; here we are essentially interested in investigating the robustness of our method for different values of β.

The examples are mainly taken from [7,4,3].

1. Wilkinson polynomial of degree 20: $\lambda(z) = \prod_{j=1}^{20}(z - j)$.
2. Monic polynomial of degree 20 with roots $[-2.1 : 0.2 : 1.7]$, in Matlab notation.
3. Monic polynomial with roots $2^k,\ k = -10, -9, \ldots, 9$.
4. Reversed Wilkinson polynomial of degree 20: $\lambda(z) = \prod_{j=1}^{20}(z - \frac{1}{j})$.
5. $\lambda(z) = 20! \sum_{j=0}^{20} \frac{z^j}{j!}$.
6. $\lambda(z) = z^{20} + z^{19} + \cdots + z + 1$.
7. Given an even positive integer N, define the complex polynomial (of degree N)

$$\lambda(z) = \prod_{k=-N/2}^{N/2-1} \left(z - \frac{2(k + 0.5) - i \sin\left(\frac{2(k+0.5)}{N-1}\right)}{N - 1} \right).$$

For comparison purposes, we also show errors for the Matlab routine `eig` applied to the companion matrix of the (unbalanced) test polynomials at points 1–5. Observe that `eig` automatically performs balancing, whereas the column "eig nobal" shows results for `eig` with the balancing feature disabled (see Tables 4–10).

These results confirm the robustness of our method under balancing.

7. Conclusion and future work

In this paper we have developed a novel implicit QR eigenvalue algorithm for companion matrices. The novel method is conceptually simpler and computationally faster than the one presented in [1], by preserving or even enhancing its numerical accuracy. In our opinion the algorithm using a compression technique is fairly optimized and a last final refinement would

Table 6
Results for polynomial 3.

	$\beta = 1$	$\beta = 2^{-2}$	$\beta = 2^{-7}$	eig	eig nobal
MaxEigAbs	1.03e−9	2.33e−10	4.19e−9	2.51e−12	3.85e−3
MaxEigRel	3.16e−7	1.19e−8	4.72e−11	4.01e−13	1.99
MaxCoeffAbs	7.19	4.96e+1	9.28e+3	6.14	11.19
MaxCoeffRel	1.57e−7	3.53e−9	9.29e−12	5.80e−14	4.69
MaxCondEig	1.38e+15	1.37e+18	4.57e+37		

Table 7
Results for polynomial 4.

	$\beta = 1$	$\beta = 2^{-3}$	$\beta = 2^{-7}$	eig	eig nobal
MaxEigAbs	1.27e−1	4.25e−4	1.47e−4	1.33e−1	1.41e−1
MaxEigRel	9.43e−1	6.16e−3	2.21e−3	9.70e−1	1.93
MaxCoeffAbs	5.44e−14	5.41e−14	5.53e−14	1.32e−14	1.40e−14
MaxCoeffRel	1.00	7.17e−14	4.55e−14	1.56	2.01e+3
MaxCondEig	2.14e+16	2.42e+15	1.88e+28		

Table 8
Results for polynomial 5.

	$\beta = 1$	$\beta = 2^{3}$	$\beta = 2^{-7}$
MaxEigAbs	5.23e−12	1.58e−11	2.56e−11
MaxEigRel	7.78e−13	2.41e−12	3.90e−12
MaxCoeffAbs	1.01e+5	1.56e+4	1.44e+6
MaxCoeffRel	1.36e−13	1.54e−14	3.82e−12
MaxCondEig	8.04e+16	1.65e+4	6.32e+56

Table 9
Results for polynomial 6. Here relative and absolute errors are the same.

	$\beta = 1$	$\beta = 2^{-7}$
MaxEigAbs	4.56e−15	1.92e+4
MaxEigRel	4.56e−15	1.92e+4
MaxCoeffAbs	1.54e−14	1.14e+23
MaxCoeffRel	1.54e−14	1.14e+23
MaxCondEig	1.38	1.03e+39

Table 10
Results for polynomial 7.

	$n = 8$	$n = 16$	$n = 32$
MaxEigAbs	3.12e−15	7.48e−14	8.75e−10
MaxEigRel	1.54e−14	8.81e−14	1.92e−8
MaxCoeffAbs	1.00e−14	8.75e−14	7.00e−12
MaxCondeig	1.97e+1	9.05e+3	4.78e+9

be to make simultaneous the double process of performing the QR iteration and reconstructing a minimal generator representation of the matrix returned as output. Experimental results are reported to show that the proposed algorithm behaves like a numerically backward stable method. The theoretical proof of backward stability is an ongoing research project. We are also planning to extend and use the compression technique for the design of fast QZ iterations for dealing with structured generalized eigenvalue problems.

Acknowledgment

The third author's work was partially supported by MIUR, grant number 20083KLJEZ.

References

[1] D.A. Bini, P. Boito, Y. Eidelman, L. Gemignani, I. Gohberg, A fast implicit QR eigenvalue algorithm for companion matrices, Linear Algebra Appl. 432 (2010) 2006–2031.
[2] D.A. Bini, F. Daddi, L. Gemignani, On the shifted QR iteration applied to companion matrices, Electron. Trans. Numer. Anal. 18 (2004) 137–152.
[3] D.A. Bini, Y. Eidelman, L. Gemignani, I. Gohberg, Fast QR eigenvalue algorithms for Hessenberg matrices which are rank-one perturbations of unitary matrices, SIAM J. Matrix Anal. Appl. 29 (2007) 566–585.
[4] S. Chandrasekaran, M. Gu, J. Xia, J. Zhu, A fast QR algorithm for companion matrices, Oper. Theory Adv. Appl. 179 (2008) 111–143.
[5] G. Golub, F. Uhlig, The QR algorithm: 50 years later its genesis by John Francis and Vera Kublanovskaya and subsequent developments, IMA J. Numer. Anal. 29 (2009) 467–485.
[6] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes. The Art of Scientific Computing, third ed., Cambridge University Press, Cambridge, 2007.
[7] M. Van Barel, R. Vandebril, P. Van Dooren, K. Frederix, Implicit double shift QR-algorithm for companion matrices, Numer. Math. 116 (2010) 177–212.
[8] R. Vandebril, M. Van Barel, N. Mastronardi, Matrix Computations and Semiseparable Matrices. Vol. II, Eigenvalue and Singular Value Methods, Johns Hopkins University Press, 2008.
[9] J. Wilkinson, The Algebraic Eigenvalue Problem, Oxford University Press, 1965.