The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order

Yuli Eidelman [2], Israel Gohberg,

School of Mathematical Sciences, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel

Vadim Olshevsky [1],

Department of Mathematics, University of Connecticut, 196 Auditorium Road, Storrs, CT 06269-3009, USA

Abstract

The QR iteration method for *tridiagonal* matrices is in the heart of one classical method to solve the general eigenvalue problem. In this paper we consider the more general class of *quasiseparable* matrices that includes not only tridiagonal but also companion, comrade, unitary Hessenberg and semiseparble matrices. A fast QR iteration method exploiting the Hermitian quasiseparable structure (and thus generalizing the classical tridiagonal scheme) is presented. The algorithm is based on an earlier work [6], and it applies to the general case of Hermitian quasiseparable matrices of an arbitrary order.

The algorithm operates on generators (i.e., a linear set of parameters defining the quasiseparable matrix), and the storage and the cost of one iteration are only linear. The results of some numerical experiments are presented.

An application of this method to solve the general eigenvalue problem via quasiseparable matrices will be analyzed separately elsewhere.

AMS classification: 15A18; 15A57; 05E35; 65F15

Key words: Eigenvalue problem, QR iteration, quasiseparable matrices, tridiagonal matrices, semiseparable matrices.

Email addresses: `eideyu@post.tau.ac.il` (Yuli Eidelman), `gohberg@post.tau.ac.il` (Israel Gohberg), `olshevsky@math.uconn.edu` (Vadim Olshevsky).

[1] Supported by the NSF grant 0242518.

[2] The draft of this paper was completed during the visit of Y.Eidelman at U Conn in September 2003.

1 Introduction

1.1. Tridiagonal, semiseparable and quasiseparable matrices.

Tridiagonal matrices are instrumental in a number of classical methods for solving the general eigenvalue problem. It is well-known [8] that the inverses of tridiagonal matrices are not sparse, but they have a $(1,1)$-semiseparable structure defined next.

A matrix A is called order (n_L, n_U)-*semiseparable* if

$$A = tril(A_L) + triu(A_U), \qquad \text{where} \qquad \text{rank}A_L = n_L, \qquad \text{rank}A_U = n_U \tag{1.1}$$

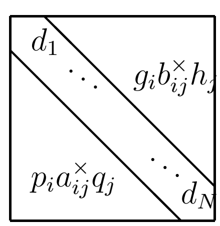
with some matrices A_L, A_U. Here $tril(A)$ and $triu(A)$ are the standard MATLAB notations standing for the strictly lower and strictly upper triangular parts of A, resp.

It is very easy to see that tridagonal matrices are not semiseparable, we will return to this observation in a moment. In fact, both belong to the more general class of *quasiseparable* matrices defined next. Following [6] we refer to A as an *order-(n_L, n_U)-quasiseparable* matrix if

$$n_L = \max \text{rank}A_{21}, \qquad n_U = \max \text{rank}A_{12}, \tag{1.2}$$

where the maximum is taken over all *symmetric* partitions of the form $A = \left[\begin{array}{c|c} * & A_{12} \\ \hline A_{21} & * \end{array}\right]$. In case $n_U = n_L = r$ one refers to A as an order-r-quasiseparable [3] matrix.

1.2. Generators. A computationally important property of $N \times N$ quasiseparable matrices A is that they can be represented by only $O((n_L + n_U)N)$ parameters via

$A =$ 

(1.3)

where $\quad a_{ij}^{\times} = a_{i-1} \cdot \ldots \cdot a_{j+1}, \quad b_{ij}^{\times} = b_{i+1} \cdot \ldots \cdot b_{j-1} \quad$ with $\quad a_{i,i+1} = b_{i,i+1} = 1.$

[3] Quasiseparable matrices are also known under different names such as matrices with a low Hankel rank [4], or weakly semiseparable matrices [11].

Here p_i $(i = 2, \dots, N)$ are vector rows and q_j $(j = 1, \dots, N-1)$ are vector columns of sizes r'_{i-1} and r'_j respectively, and a_k $(k = 2, \dots, N-1)$ are matrices of sizes $r'_k \times r'_{k-1}$;these elements are said to be *lower generators of the matrix* A with *orders* r'_k $(k = 1, \dots, N-1)$. The elements g_i $(i = 1, \dots, N-1)$ are vector rows and h_j $(j = 2, \dots, N)$ are vector columns of sizes r''_i and r''_{j-1} respectively, and b_k $(k = 2, \dots, N-1)$ are matrices of sizes $r''_{k-1} \times r''_k$; these elements are said to be *upper generators of the matrix* A with *orders* r''_k, $(k = 1, \dots, N-1)$. The numbers d_k $(k = 1, \dots, N)$ are the *diagonal entries* of the matrix A. Set $n_L = \max_{1 \le k \le N-1} r'_k$, $n_U = \max_{1 \le k \le N-1} r''_k$, the matrix R is said to be *lower quasiseparable* of order n_L and *upper quasiseparable* of order n_U or *quasiseparable* of order (n_L, n_U). Though we focus in this paper on algorithms for matrices with scalar entries, block quasiseparable matrices are also used in our proofs and computations.

1.3. The tridiagonal and quasiseparable QR iteration. It is well known that the QR iteration can be implemented very efficiently for tridiagonal A. The relevant facts are: (i) $A = QR$ can be computed cheaply; (ii) the next iterant $A_1 = RQ$ is also tridiagonal [13]. Both facts carry over to the more general Hermitian quasiseparable case: (i) we can also compute $A = QR$ fast (cf with [6]); (ii) the next iterant $A_1 = RQ$ has the same order of quasiseparability (cf with [5]). Moreover, as explained next the algorithms for computing Q, R, and $A_1 = RQ$ can be constructed from building blocks that have already been used in [5,6] for different purposes.

1.4. The structure of the Q and R factors. To compute the generators of the Q and R factors we adapt the algorithm of [6] (derived there via applying the more general Dewilde-Van der Veen method [4] to *finite* quasiseparable matrices). Specifically, given generators of an order (n_L, n_U) quasiseparable matrix A, the algorithm of [6] computes

$$\underbrace{A}_{(n_L,n_U)} = \underbrace{V}_{(n_L,*)} \cdot \underbrace{U}_{(*,n_L)} \cdot \underbrace{R}_{(0,n_L+n_U)}$$

where the order of quasiseparability is shown in the bottom line. Here V, U are unitary matrices and R is an upper triangular matrix.

Matrices V and U are quasiseparable, so it is not surprising that $Q = VU$ has some structure as well. It can be shown that the factor $Q = VU$ is order (n_L, n_L) quasiseparable. Hence in order to convert an algorithm of [6] into a form computing the generators of Q and R one just needs to provide a structure-exploiting method for multiplying V and U. Such multiplication of quasiseparable matrices has already been used in [5], and here we adapt it to produce an $O(N)$ algorithm for computing generators of the factors in

$$\underbrace{A}_{(n_L,n_U)} = \underbrace{Q}_{(n_L,n_L)} \cdot \underbrace{R.}_{(0,n_L+n_U)} \tag{1.4}$$

1.4. Quasiseparable QR iteration. If A is Hermitian, then (1.4) becomes

$$\underbrace{A}_{(n,n)} = \underbrace{Q}_{(n,n)} \cdot \underbrace{R.}_{(0,2n)} \tag{1.5}$$

Now, observe that the Hermitian quasiseparable structure is inherited under QR iteration, i.e., that the order of quasiseparability of A_1 is also n:

$$\underbrace{A_1}_{(n,n)} = \underbrace{R}_{(0,2n)} \cdot \underbrace{Q.}_{(n,n)} \tag{1.6}$$

Though this fact follows from the multiplication algorithms of [5] (applied to the particular patterns of Q and R shown in (1.6)), we provide here a different and a more transparent explanation. Let us partition A and A_1 as in

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \cdot \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}$$

and

$$A_1 = \begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} \\ A_{21}^{(1)} & A_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \cdot \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix},$$

so that $A_{21} = Q_{21}R_{11}$ and $A_{21}^{(1)} = R_{22}Q_{21}$. Assume, for simplicity, that if R is invertible (this assumption is not used in the paper). Then it is immediate to see that $rank A_{21} = rank A_{21}^{(1)}$. In the Hermitian case it means that A_1 inherits the order of quasiseparability of A. Again, the techniques of [5] for multiplication of quasiseparable matrices can be adapted to compute $A_1 = RQ$, or A_1 for the shifted QR iteration method

$$\begin{cases} A - \sigma I = QR, \\ A_1 = \sigma I + RQ, \end{cases}$$

1.5. The context and the structure of the paper.

Various numerical methods for quasiseparable matrices are being suggested in parallel papers [1–3,7,9,10,12]. Their QR-iteration methods were studied for several special cases of quasi-separable matrices, mostly for the order-one matrices and their derivatives. For example, a rank-one perturbation of a symmetric order-one matrix is considered in [1]. A symmetric order-one diagonal-plus-semiseparable matrix is studied in [12]. Remark that tridiaginal matrices are not semiseparable and hence the semiseparable algorithms are not generalizations of the classical methods rather their alternatives.

The algorithms presented here differ from their in three aspects. First, they are not limited to the low *order of semiseparability*. Secondly, they apply to the more general Hermitian *quasiseparable* structure and hence generalize both classical tridiagonal and semi-separable methods. Finally, they are based on an the earlier work and *explicitly* employ the ingredients available in the literature. An application of our method to solve the general eigenvalue problem via quasiseparable matrices will appear separately elsewhere.

The paper consists of seven sections. Section 1 is the introduction. Section 2 contains the definitions. In Section 3 we present without proof an algorithm which is a modification of the Dewilde-Van der Veen method and based on this algorithm derive an algorithm for QR factorization of a quasiseparable matrix. In Section 4 we consider one step of QR iteration for quasiseparable matrices and derive an algorithm to compute the lower generators and the diagonal entries of the next iterant via the generators of the previous one. In Section 5 we consider the case of a Hermitian quasiseparable matrix. Section 6 contains expressions for complexities of the obtained algorithms. In Section 7 we presents results of numerical experiments.

2 Definitions

Let $\{a_k\}, k = 1, \ldots, N$ be a family of matrices of sizes $r_k \times r_{k-1}$. For positive integers $i, j,\ i > j$ define the operation $a^{\times}_{ij}$ as follows: $a^{\times}_{ij} = a_{i-1} \cdots a_{j+1}$ for $i > j+1$, $a^{\times}_{j+1,j} = I_{r_j}$.

Let $\{b_k\}, k = 1, \ldots, N$ be a family of matrices of sizes $r_{k-1} \times r_k$. For positive integers $i, j,\ j > i$ define the operation $b^{\times}_{ij}$ as follows: $b^{\times}_{ij} = b_{i+1} \cdots b_{j-1}$ for $j > i+1$, $b^{\times}_{i,i+1} = I_{r_i}$.

It is easy to see that

$$a^{\times}_{ik} = a^{\times}_{ij} a^{\times}_{j+1,k}, \quad i > j \geq k \tag{2.1}$$

and

$$b^{\times}_{kj} = b^{\times}_{k,i+1} b^{\times}_{i,j}, \quad k \leq i < j. \tag{2.2}$$

Let $R = \{R_{ij}\}_{i,j=1}^{N}$ be a matrix with block entries R_{ij} of sizes $m_i \times n_j$. Assume that the entries of this matrix are represented in the form

$$R_{ij} = \begin{cases} p_i a^{\times}_{ij} q_j, & 1 \leq j < i \leq N, \\ d_i, & 1 \leq i = j \leq N, \\ g_i b^{\times}_{ij} h_j, & 1 \leq i < j \leq N. \end{cases} \tag{2.3}$$

Here p_i $(i=2,\dots,N)$, q_j $(j=1,\dots,N-1)$, a_k $(k=2,\dots,N-1)$ are matrices of sizes $m_i \times r'_{i-1}$, $r'_j \times n_j$, $r'_k \times r'_{k-1}$ respectively; these elements are said to be *lower generators of the matrix* R with *orders* r'_k $(k=1,\dots,N-1)$. The elements g_i $(i=1,\dots,N-1)$, h_j $(j=2,\dots,N)$, b_k $(k=2,\dots,N-1)$ are matrices of sizes $m_i \times r''_i$, $r''_{j-1} \times n_j$, $r''_{k-1} \times r''_k$ respectively; these elements are said to be *upper generators of the matrix* R with *orders* r''_k, $(k=1,\dots,N-1)$. The matrices d_k $(k=1,\dots,N)$ of sizes $m_k \times n_k$ are said to be *diagonal entries* of the matrix R. We define also orders of generators r'_k, r''_k for $k=0$, N setting them to be zeros. For scalar matrices the generators p_i, g_i and q_j, h_j are rows and columns of the corresponding sizes. Set $n_L = \max_{1\le k\le N-1} r'_k$, $n_U = \max_{1\le k\le N-1} r''_k$, the matrix R is said to be *lower quasiseparable* of order n_L and *upper quasiseparable* of order n_U or *quasiseparable* of order (n_L, n_U).

Formally, we use some calculation rules with matrices that have blocks with dimension zero. Aside from obvious rules, the product of an "empty" matrix of dimension $m \times 0$ and an empty matrix of dimension $0 \times n$ is a matrix of dimension $m \times n$ with all elements equal to 0. All further rules of block matrix multiplication remain consistent. Such operations are used in MATLAB.

3 The QR factorization

Let $R = \{R_{ij}\}_{i,j=1}^N$ be a matrix with entries from $\mathbb{C}$ with given generators. We present here an algorithm for computing generators and diagonal entries of unitary matrix Q and upper triangular matrix S such that $R = QS$. The main part of the algorithm is based on the following result from [6].

Theorem 3.1 *Let $R = \{R_{ij}\}_{i,j=1}^N$ be a scalar matrix with lower generators p_i $(i=2,\dots,N)$, q_j $(j=1,\dots,N-1)$, a_k $(k=2,\dots,N-1)$ of orders r'_k $(k=1,\dots,N-1)$, upper generators g_i $(i=1,\dots,N-1)$, h_j $(j=2,\dots,N)$, b_k $(k=2,\dots,N-1)$ of orders r''_k $(k=1,\dots,N-1)$ and diagonal entries d_k $(k=1,\dots,N)$. Let us define the numbers ρ_k via recursive relations $\rho_N = 0$, $\rho_{k-1} = \min\{1+\rho_k,\ r'_{k-1}\}$, $k=N,\dots,2$, $\rho_0 = 0$ and the numbers $m_k = 1, n_k = 1, \nu_k = 1 + \rho_k - \rho_{k-1}$, $\rho'_k = r''_k + \rho_k$, $k = 1,\dots,N$.*

The matrix R admits the factorization

$$R = VUS,$$

where V is a unitary matrix represented in the block lower triangular form with blocks of sizes $m_i \times \nu_j$ $(i,j=1,\dots,N)$, lower generators $(p_V)_i$ $(i=2,\dots,N)$, $(q_V)_j$ $(j=1,\dots,N-1)$, $(a_V)_k$ $(k=2,\dots,N-1)$ of orders ρ_k $(k=1,\dots,N-1)$ and diagonal entries $(d_V)_k$ $(k=1,\dots,N)$, U is a unitary matrix represented in the block upper triangular form with blocks of sizes $\nu_i \times$

n_j $(i,j = 1,\dots,N)$, upper generators $(g_U)_i$ $(i = 1,\dots,N-1)$, $(h_U)_j$ $(j = 2,\dots,N)$, $(b_U)_k$ $(k = 2,\dots,N-1)$ of orders ρ_k $(k = 1,\dots,N-1)$ and diagonal entries $(d_U)_k$ $(k = 1,\dots,N)$ and S is an upper triangular matrix with upper generators $(g_S)_i$ $(i = 1,\dots,N-1)$, $(h_S)_j$ $(j = 2,\dots,N)$, $(b_S)_k$ $(k = 2,\dots,N-1)$ of orders ρ'_k $(k = 1,\dots,N-1)$ and diagonal entries $(d_S)_k$ $(k = 1,\dots,N)$.

The generators and the diagonal entries of the matrices V, U, S are determined using the following algorithm.

1.1. If $r'_{N-1} > 0$ set

$$X_N = p_N, \quad (p_V)_N = 1, \quad (h_S)_N = \begin{bmatrix} h_N \\ d_N \end{bmatrix},$$

$(d_V)_N$ to be 1×0 empty matrix, Δ_N to be 0×1 empty matrix;

if $r'_{N-1} = 0$ set X_N to be 0×0 empty matrix, $(p_V)_N$ to be 1×0 empty matrix,

$$(d_V)_N = 1, \quad (h_S)_N = h_N, \quad \Delta_N = d_N.$$

1.2. For $k = N-1,\dots,2$ perform the following. Compute the QR factorization

$$\begin{bmatrix} p_k \\ X_{k+1}a_k \end{bmatrix} = V_k \begin{pmatrix} X_k \\ 0 \end{pmatrix},$$

where V_k is a unitary matrix of sizes $(1+\rho_k)\times(1+\rho_k)$, X_k is a matrix of sizes $\rho_{k-1}\times r'_{k-1}$. Determine matrices $(p_V)_k$, $(a_V)_k$, $(d_V)_k$, $(q_V)_k$ of sizes $1\times\rho_{k-1}$, $\rho_k\times\rho_{k-1}$, $1\times\nu_k$, $\rho_k\times\nu_k$ from the partition

$$V_k = \begin{bmatrix} (p_V)_k & (d_V)_k \\ (a_V)_k & (q_V)_k \end{bmatrix}.$$

Compute

$$h'_k = (p_V)_k^* d_k + (a_V)_k^* X_{k+1} q_k, \quad (h_S)_k = \begin{bmatrix} h_k \\ h'_k \end{bmatrix}, \quad (b_S)_k = \begin{pmatrix} b_k & 0 \\ (p_V^*)_k g_k & (a_V)_k^* \end{pmatrix},$$
$$\Theta_k = \begin{bmatrix} (d_V)_k^* g_k & (q_V)_k^* \end{bmatrix}, \quad \Delta_k = (d_V)_k^* d_k + (q_V)_k^* X_{k+1} q_k.$$

1.3. Set $V_1 = I_{\nu_1}$ and define matrices $(d_V)_1$, $(q_V)_1$ of sizes $1\times\nu_1$, $\rho_1\times\nu_1$ from *the partition*

$$V_1 = \begin{bmatrix} (d_V)_1 \\ (q_V)_1 \end{bmatrix};$$

compute

$$\Delta_1 = \begin{pmatrix} d_1 \\ X_2 q_1 \end{pmatrix}, \; \Theta_1 = \begin{pmatrix} g_1 & 0 \\ 0 & I_{\rho_1} \end{pmatrix}.$$

Thus we have computed generators and diagonal entries of the matrix V and generators $(b_S)_k$, $(h_S)_k$ of the matrix S.

2.1. Compute the QR factorization

$$\begin{bmatrix} \Delta_1 & \Theta_1 \end{bmatrix} = U_1 \begin{bmatrix} (d_S)_1 & (g_S)_1 \\ 0 & Y_1 \end{bmatrix},$$

where U_1 is a unitary matrix of sizes $\nu_1 \times \nu_1$, $(d_S)_1$ is a number, $(g_S)_1$ is a row of size ρ'_1, Y_1 is a matrix of sizes $\rho_1 \times \rho'_1$. Determine matrices $(d_U)_1$, $(g_U)_1$ of sizes $\nu_1 \times 1$, $\nu_1 \times \rho'_1$ from the partition

$$U_1 = \begin{bmatrix} (d_U)_1 & (g_U)_1 \end{bmatrix}.$$

2.2. For $k = 2, \ldots, N-1$ perform the following. Compute the QR factorization

$$\begin{bmatrix} Y_{k-1}(h_S)_k & Y_{k-1}(b_S)_k \\ \Delta_k & \Theta_k \end{bmatrix} = U_k \begin{bmatrix} (d_S)_k & (g_S)_k \\ 0 & Y_k \end{bmatrix},$$

where U_k is a unitary matrix of sizes $(1+\rho_k) \times (1+\rho_k)$, $(d_S)_k$ is a number, $(g_S)_k$ is a row of size ρ'_k, Y_k is a matrix of sizes $\rho_k \times \rho'_k$. Determine matrices $(d_U)_k$, $(g_U)_k$, $(h_U)_k$, $(b_U)_k$ of sizes $(1+\rho_k-\rho_{k-1}) \times 1$, $(1+\rho_k-\rho_{k-1}) \times \rho_k$, $\rho_{k-1} \times 1$, $\rho_{k-1} \times \rho_k$ from the partition

$$U_k = \begin{bmatrix} (h_U)_k & (b_U)_k \\ (d_U)_k & (g_U)_k \end{bmatrix}.$$

2.3. If $r'_{N-1} > 0$ set $(h_U)_N = 1$ and $(d_U)_N$ to be 0×1 empty matrix;

if $r'_{N-1} = 0$ set $(d_U)_N = 1$ and $(h_U)_N$ to be 0×1 empty matrix;

compute

$$(d_S)_N = \begin{bmatrix} Y_{N-1}(h_S)_N \\ \Delta_N \end{bmatrix}.$$

Thus we have computed generators and diagonal entries of the matrix U and generators $(g_S)_k$ and diagonal entries $(d_S)_k$ of the matrix S.

Theorem 3.1 yields the QR-factorization of the matrix R, i.e. representation of R in the form $R = QS$ with the unitary matrix $Q = UV$ and the upper triangular matrix S. For the next considerations we should obtain generators of the matrix Q explicitly.

Theorem 3.2 *Let $R = \{R_{ij}\}_{i,j=1}^N$ be a scalar matrix with lower generators p_i $(i = 2,\dots,N)$, q_j $(j = 1,\dots,N-1)$, a_k $(k = 2,\dots,N-1)$ of orders r'_k $(k = 1,\dots,N-1)$, upper generators g_i $(i = 1,\dots,N-1)$, h_j $(j = 2,\dots,N)$, b_k $(k = 2,\dots,N-1)$ of orders r''_k $(k = 1,\dots,N-1)$ and diagonal entries d_k $(k = 1,\dots,N)$. Let us define the numbers ρ_k via recursive relations $\rho_N = 0$, $\rho_{k-1} = \min\{1+\rho_k,\ r'_{k-1}\}$, $k = N,\dots,2$, $\rho_0 = 0$ and the numbers $\rho'_k = r''_k + \rho_k$, $k = 1,\dots,N$.*

The matrix R admits the factorization

$$R = QS,$$

where Q is a unitary matrix with lower generators $(p_Q)_i$ $(i = 2,\dots,N)$, $(q_Q)_j$ $(j = 1,\dots,N-1)$, $(a_Q)_k$ $(k = 2,\dots,N-1)$ of orders ρ_k $(k = 1,\dots,N-1)$, upper generators $(g_Q)_i$ $(i = 1,\dots,N-1)$, $(h_Q)_j$ $(j = 2,\dots,N)$, $(b_Q)_k$ $(k = 2,\dots,N-1)$ of orders ρ_k $(k = 1,\dots,N-1)$ also and diagonal entries $(d_Q)_k$ $(k = 1,\dots,N)$ and S is an upper triangular matrix with upper generators $(g_S)_i$ $(i = 1,\dots,N-1)$, $(h_S)_j$ $(j = 2,\dots,N)$, $(b_S)_k$ $(k = 2,\dots,N-1)$ of orders ρ'_k $(k = 1,\dots,N-1)$ and diagonal entries $(d_S)_k$ $(k = 1,\dots,N)$.

The generators and the diagonal entries of the matrices Q and S are determined using the following algorithm.

1. Using the algorithm from Theorem 3.1 compute generators and diagonal entries of the upper triangular matrix S and of the unitary block triangular matrices V and U such that $R = VUS$.

2. Compute generators and diagonal entries of the matrix $Q = VU$ using generators and diagonal entries of the matrices V, U as follows.

2.1. Compute

$$z_1 = (q_V)_1(g_U)_1,$$
$$(q_Q)_1 = (q_V)_1(d_U)_1, \quad \alpha_1 = (a_V)_2 z_1, \tag{3.1}$$

$$(d_Q)_1 = (d_V)_1(d_U)_1, \quad \beta_1 = z_1, \tag{3.2}$$

$$(g_Q)_1 = (d_V)_1(g_U)_1, \quad \gamma_1 = z_1(b_U)_2. \tag{3.3}$$

Set $(a_V)_N = 0_{0\times\rho_{N-1}}$, $(b_V)_N = 0_{\rho_{N-1}\times 0}$.

2.2. For $i = 2, \dots, N-1$ *perform the following. Set*

$$(p_Q)_i = (p_V)_i, \quad (a_Q)_i = (a_V)_i, \quad (b_Q)_i = (b_U)_i, \quad (h_Q)_i = (h_U)_i.$$

Compute

$$z_i = (q_V)_i(g_U)_i, \tag{3.4}$$

$$(q_Q)_i = (q_V)_i(d_U)_i + \alpha_{i-1}(h_U)_i, \quad \alpha_i = (a_V)_{i+1}[z_i + \alpha_{i-1}(b_U)_i], \tag{3.5}$$

$$(d_Q)_i = (d_V)_i(d_U)_i + (p_V)_i\beta_{i-1}(h_U)_i, \quad \beta_i = z_i + (a_V)_i\beta_{i-1}(b_U)_i, \tag{3.6}$$

$$(g_Q)_i = (d_V)_i(g_U)_i + (p_V)_i\gamma_{i-1}, \quad \gamma_i = [z_i + (a_V)_i\gamma_{i-1}](b_U)_{i+1}. \tag{3.7}$$

2.3. Set $(p_Q)_N = (p_V)_N$, $(h_Q)_N = (h_U)_N$. *Compute*

$$(d_Q)_N = (d_V)_N(d_U)_N + (p_V)_N\beta_{N-1}(h_U)_N. \tag{3.8}$$

Proof. We should justify the second stage of the algorithm. Let $Q = \{Q_{ij}\}_{i,j=1}^N$, $V = \{V_{ij}\}_{i,j=1}^N$, $U = \{U_{ij}\}_{i,j=1}^N$. For $N \geq i > j \geq 1$ since U is an upper triangular matrix and $(p_V)_i$ $(i = 2, \dots, N)$, $(q_V)_j$ $(j = 1, \dots, N-1)$, $(a_V)_k$ $(k = 2, \dots, N-1)$ are lower generators of the matrix V we have

$$Q_{ij} = \sum_{k=1}^{j} V_{ik}U_{kj} = \sum_{k=1}^{j} (p_V)_i(a_V)^{\times}_{ik}(q_V)_k U_{kj}.$$

Using the equality (2.1) we obtain

$$Q_{ij} = (p_V)_i(a_V)^{\times}_{ij}(q_Q)_j, \quad 1 \leq j < i \leq N$$

where

$$(q_Q)_j = \sum_{k=1}^{j} (a_V)^{\times}_{j+1,k}(q_V)_k U_{kj}, \quad j = 1, \dots, N-1. \tag{3.9}$$

This implies that the matrix Q has the lower generators $(p_Q)_i = (p_V)_i$ $(i = 2, \dots, N)$, $(a_Q)_k = (a_V)_k$ $(k = 2, \dots, N-1)$ and $(q_Q)_j$ $(j = 1, \dots, N-1)$ defined in (3.9). This in particular means that the orders ρ_k $(k = 1, \dots, N-1)$ of these generators are the same as for the matrix V. Now we must check that the generators $(q_Q)_j$ satisfy the relations (3.1), (3.5). Indeed for $j = 1$ we have

$$(q_Q)_1 = (a_V)^{\times}_{2,1}(q_V)_1 U_{11} = (q_V)_1(d_U)_1$$

and for $j = 2, \dots, N-1$ using $U_{jj} = (d_U)_j$ and the fact that $(g_U)_i$ $(i = 1, \dots, N-1)$, $(h_U)_j$ $(j = 2, \dots, N)$, $(b_U)_k$ $(k = 2, \dots, N-1)$ are the upper generators of the matrix U we get

$$(q_Q)_j = \sum_{k=1}^{j-1}(a_V)^{\times}_{j+1,k}(q_V)_k(g_U)_k(b_U)^{\times}_{kj}(h_U)_j+(a_V)^{\times}_{j+1,j}(q_V)_j(d_U)_j = \alpha_{j-1}(h_U)_j+(q_V)_j(d_U)_j,$$

where

$$\alpha_{j-1} = \sum_{k=1}^{j-1}(a_V)^{\times}_{j+1,k}(q_V)_k(g_U)_k(b_U)^{\times}_{kj}.$$

We have

$$\alpha_1 = (a_V)^{\times}_{3,1}(q_V)_1(g_U)_1(b_U)^{\times}_{2,1} = (a_V)_2(q_V)_1(g_U)_1$$

and using the relations (2.1), (2.2) we obtain

$$\alpha_j = \sum_{k=1}^{j}(a_V)^{\times}_{j+2,k}(q_V)_k(g_U)_k(b_U)^{\times}_{k,j+1}$$

$$= (a_V)^{\times}_{j+2,j}(q_V)_j(g_U)_j(b_U)^{\times}_{j,j+1} + (a_V)_{j+1}(\sum_{k=1}^{j-1}(a_V)^{\times}_{j+1,k}(q_V)_k(g_U)_k(b_U)^{\times}_{kj})(b_U)_j$$

$$= (a_V)_{j+1}(q_V)_j(g_U)_j + (a_V)_{j+1}\alpha_{j-1})(b_U)_j$$

which completes the proof of (3.1), (3.5).

For diagonal entries of the matrix Q we have

$$(d_Q)_1 = Q_{11} = V_{11}U_{11} = (d_V)_1(d_U)_1$$

and for $i = 2, \ldots, N$

$$Q_{ii} = \sum_{k=1}^{i} V_{ik}U_{ki} = V_{ii}U_{ii} + \sum_{k=1}^{i-1} V_{ik}U_{ki} = (d_V)_i(d_U)_i + (p_V)_i\beta_{i-1}(h_U)_i,$$

where

$$\beta_{i-1} = \sum_{k=1}^{i-1}(a_V)^{\times}_{ik}(q_V)_k(g_U)_k(b_U)^{\times}_{ki}$$

We have $\beta_1 = (q_V)_1(g_U)_1$ and using the relations (2.1), (2.2) we obtain

$$\beta_i = \sum_{k=1}^{i}(a_V)^{\times}_{i+1,k}(q_V)_k(g_U)_k(b_U)^{\times}_{k,i+1}$$

$$= (a_V)^{\times}_{i+1,i}(q_V)_i(g_U)_i(b_U)^{\times}_{i,i+1} + (a_V)_i(\sum_{k=1}^{j-1}(a_V)^{\times}_{ik}(q_V)_k(g_U)_k(b_U)^{\times}_{ki})(b_U)_i$$

$$= (q_V)_i(g_U)_i + (a_V)_i\beta_{i-1})(b_U)_i$$

which completes the proof of (3.2), (3.6), (3.8).

The proof of the relations (3.3), (3.7) is performed in the same way as the proof of (3.1), (3.5). $\square$

Corollary 1 *Let R be a quasiseparable of order (n_L, n_U) matrix with scalar entries and let $R = QS$ be the factorization obtained in Theorem 3.2. Then the unitary matrix Q is quasiseparable of order (n_L, n_L) at most and the upper triangular matrix S is upper quasiseparable of order $n_L + n_U$ at most.*

Proof. By Theorem 3.2 the matrix Q has lower and upper generators of the orders ρ_k $(k = 1, \ldots, N-1)$ defined by the relations

$$\rho_N = 0,\ \rho_{k-1} = \min\{1 + \rho_k,\ r'_{k-1}\},\ k = N, \ldots, 2 \tag{3.10}$$

and by Theorem 3.1 the matrix S has upper generators of orders

$$\rho'_k = r''_k + \rho_k,\ k = 1, \ldots, N-1. \tag{3.11}$$

From the inequalities $r'_k \leq n_L$ $(k = 1, \ldots, N-1)$ and the relations (3.10) it follows that

$$\rho_k \leq r'_k \leq n_L, \quad k = 1, \ldots, N-1 \tag{3.12}$$

and hence the maximal order of generators of the matrix Q is not greater than n_L. Next from (3.11) and (3.12) we conclude that the maximal order of upper generators of the matrix S is not greater than $n_L + n_U$. $\square$

4 The QR iteration

We consider the QR iteration algorithm for matrices defined via generators. In each iteration step for a given matrix R and for a given real number σ the new iterant R_1 is obtained by the rule

$$\begin{cases} R - \sigma I = QS, \\ R_1 = \sigma I + SQ, \end{cases}$$

where Q is a unitary matrix and S is an upper triangular matrix. We show that the matrix R_1 has lower generators with the same orders as for the lower generators of the matrix Q and hence these orders are not greater than the orders of the corresponding generators of the matrix R and obtain an algorithm for computation of these generators and the diagonal entries of the matrix R_1.

Theorem 4.1 *Let $R = \{R_{ij}\}_{i,j=1}^N$ be a scalar matrix with lower generators p_i $(i = 2, \ldots, N)$, q_j $(j = 1, \ldots, N-1)$, a_k $(k = 2, \ldots, N-1)$ of orders r'_k $(k = 1, \ldots, N-1)$, upper generators g_i $(i = 1, \ldots, N-1)$, h_j $(j = 2, \ldots, N)$, b_k $(k = 2, \ldots, N-1)$ of orders r''_k $(k = 1, \ldots, N-1)$ and diagonal entries d_k $(k = 1, \ldots, N)$ and σ be a real number. Let us define the numbers ρ_k via recursive relations $\rho_N = 0,\ \rho_{k-1} = \min\{1 + \rho_k,\ r'_{k-1}\},\ k = N, \ldots, 2,\ \rho_0 =$*

0. *Define the matrix* R_1 *by the rule*

$$\begin{cases} R - \sigma I = QS, \\ R_1 = \sigma I + SQ, \end{cases}$$

where Q *is a unitary matrix and* S *is an upper triangular matrix.*

The matrix R_1 *has lower generators of orders* ρ_k $(k = 1, \dots, N-1)$. *These lower generators* $p_i^{(1)}$ $(i = 2, \dots, N)$, $q_j^{(1)}$ $(j = 1, \dots, N-1)$, $a_k^{(1)}$ $(k = 2, \dots, N-1)$ *and the diagonal entries* $d_k^{(1)}$ $(k = 1, \dots, N)$ *of the matrix* R *are determined using the following algorithm.*

1. *Apply to the matrix* $R - \sigma I$, *which has the same lower and upper generators as the matrix* R *and the diagonal entries* $d_k - \sigma$ $(k = 1, \dots, N)$, *the algorithm from Theorem 3.2, to compute the lower generators* $(p_Q)_i$ $(i = 2, \dots, N)$, $(q_Q)_j$ $(j = 1, \dots, N-1)$, $(a_Q)_k$ $(k = 2, \dots, N-1)$ *and the diagonal entries* $(d_Q)_k$ $(k = 1, \dots, N)$ *of the matrix* Q *and the upper generators* $(g_S)_i$ $(i = 1, \dots, N-1)$, $(h_S)_j$ $(j = 2, \dots, N)$, $(b_S)_k$ $(k = 2, \dots, N-1)$ *and the diagonal entries* $(d_S)_k$ $(k = 1, \dots, N)$ *of the matrix* S.

2. *Compute the lower generators and the diagonal entries of the matrix* R_1 *as follows.*

2.1. *Compute*

$$z_N = (h_S)_N (p_Q)_N,$$

$$p_N^{(1)} = (d_S)_N (p_Q)_N, \quad \alpha_N = z_N (a_Q)_{N-1}, \tag{4.1}$$

$$d_N^{(1)} = (d_S)_N (d_Q)_N + \sigma, \quad \beta_N = z_N. \tag{4.2}$$

Set $(a_Q)_1 = 0_{\rho_1 \times 0}$.

2.2. *For* $i = N-1, \dots, 2$ *perform the following. Set*

$$q_i^{(1)} = (q_Q)_i, \quad a_i^{(1)} = (a_Q)_i.$$

Compute

$$z_i = (h_S)_i (p_Q)_i, \tag{4.3}$$

$$p_i^{(1)} = (d_S)_i (p_Q)_i + (g_S)_i \alpha_{i+1}, \quad \alpha_i = [z_i + (b_S)_i \alpha_{i+1}](a_Q)_{i-1}, \tag{4.4}$$

$$d_i^{(1)} = (d_S)_i (d_Q)_i + (g_S)_i \beta_{i+1} (q_Q)_i + \sigma, \quad \beta_i = z_i + (b_S)_i \beta_{i+1} (a_Q)_i. \tag{4.5}$$

2.3. *Set* $q_1^{(1)} = (q_Q)_1$. *Compute*

$$d_1^{(1)} = (d_S)_1 (d_Q)_1 + (g_S)_1 \beta_2 (q_Q)_1 + \sigma. \tag{4.6}$$

Proof. We should justify the second stage of the algorithm. Let $Q = \{Q_{ij}\}_{i,j=1}^N$, $S = \{S_{ij}\}_{i,j=1}^N$ and $R_1 = \{R_{ij}^{(1)}\}_{i,j=1}^N$. For $N \geq i > j \geq 1$ using the fact S is an upper triangular matrix and $(p_Q)_i$ $(i = 2, \dots, N)$, $(q_Q)_j$ $(j = 1, \dots, N-1)$, $(a_Q)_k$ $(k = 2, \dots, N-1)$ are lower generators of the matrix Q we have

$$R_{ij}^{(1)} = \sum_{k=i}^N S_{ik}Q_{kj} = \sum_{k=i}^N S_{ik}(p_Q)_k(a_Q)_{kj}^{\times}(q_Q)_j.$$

Using the equality (2.1) we obtain

$$R_{ij}^{(1)} = p_i^{(1)}(a_Q)_{ij}^{\times}(q_Q)_j, \quad 1 \leq j < i \leq N$$

where

$$p_i^{(1)} = \sum_{k=i}^N S_{ik}(p_Q)_k(a_Q)_{k,i-1}^{\times}, \quad i = 2, \dots, N. \tag{4.7}$$

This implies that the matrix $R^{(1)}$ has the lower generators $a_k^{(1)} = (a_Q)_k$ $(k = 2, \dots, N-1)$, $q_j^{(1)} = (q_Q)_j$ $(j = 1, \dots, N-1)$ and $p^{(1)}$ $(i = 2, \dots, N)$ defined in (4.7). This in particular means that the orders ρ_k $(k = 1, \dots, N-1)$ of these generators are the same as for the matrix Q. Now we must check that the generators $p_i^{(1)}$ satisfy the relations (4.1), (4.4). Indeed for $i = N$ we have

$$p_N^{(1)} = S_{NN}(p_Q)_N(a_Q)_{N,N-1}^{\times} = (d_S)_N(p_Q)_N$$

and for $i = N-1, \dots, 2$ using $S_{jj} = (d_S)_j$ and the fact that $(g_S)_i$ $(i = 1, \dots, N-1)$, $(h_S)_j$ $(j = 2, \dots, N)$, $(b_S)_k$ $(k = 2, \dots, N-1)$ are the upper generators of the matrix S we get

$$p_i^{(1)} = (g_S)_i \sum_{k=i+1}^N (b_S)_{ik}^{\times}(h_S)_k(p_Q)_k(a_Q)_{k,i-1}^{\times} + (d_S)_i(p_Q)_i(a_Q)_{i,i-1}^{\times} = (d_S)_i(p_Q)_i + (g_S)_i\alpha_{i+1},$$

where

$$\alpha_{i+1} = \sum_{k=i+1}^N (b_S)_{ik}^{\times}(h_S)_k(p_Q)_k(a_Q)_{k,i-1}^{\times}.$$

We have

$$\alpha_N = (b_S)_{N-1,N}^{\times}(h_S)_N(p_Q)_N(a_Q)_{N,N-2}^{\times} = (h_S)_N(p_Q)_N(a_Q)_{N-1}$$

and using the relations (2.1), (2.2) we obtain

$$\alpha_i = \sum_{k=i}^N (b_S)_{i-1,k}^{\times}(h_S)_k(p_Q)_k(a_Q)_{k,i-2}^{\times}$$

$$= (b_S)_{i-1,i}^{\times}(h_S)_i(p_Q)_i(a_Q)_{i,i-2}^{\times} + (b_S)_i\Big(\sum_{k=i+1}^N (b_S)_{ik}^{\times}(h_S)_k(p_Q)_k(a_Q)_{k,i-1}^{\times}\Big)(a_Q)_{i-1}$$

$$= [(h_S)_i(p_Q)_i + (b_S)_i\alpha_{i+1}](a_Q)_{i-1}$$

which completes the proof of (4.1), (4.4).

For diagonal entries of the matrix S we have

$$d_N^{(1)} = R_{NN}^{(1)} = S_{NN}Q_{NN} = (d_S)_N(d_Q)_N$$

and for $i = N-1, \dots, 1$

$$R_{ii}^{(1)} = \sum_{k=i}^{N} S_{ik}Q_{ki} = S_{ii}Q_{ii} + \sum_{k=i+1}^{N} S_{ik}Q_{ki} = (d_S)_i(d_Q)_i + (g_S)_i\beta_{i+1}(h_S)_i,$$

where

$$\beta_{i+1} = \sum_{k=i+1}^{N} (b_S)_{ik}^{\times}(h_S)_k(p_Q)_k(a_Q)_{ki}^{\times}$$

We have $\beta_1 = (q_V)_1(g_U)_1$ and using the relations (2.1), (2.2) we obtain

$$\beta_i = \sum_{k=i}^{N}(b_S)_{i-1,k}^{\times}(h_S)_k(p_Q)_k(a_Q)_{k,i-1}^{\times}$$

$$= (b_S)_{i-1,i}^{\times}(h_S)_i(p_Q)_i(a_Q)_{i,i-1}^{\times} + (b_S)_i(\sum_{k=i+1}^{N}(b_S)_{ik}^{\times}(h_S)_k(p_Q)_k(a_Q)_{ki}^{\times})(a_Q)_i =$$

$$(h_S)_i(p_Q)_i + (b_S)_i\beta_{i+1})(a_Q)_i$$

which completes the proof of (4.2), (4.5), (4.6). $\square$

Corollary 2 *Let R be a lower quasiseparable of order n_L matrix with scalar entries and let R_1 be the matrix obtained in Theorem 3.2. Then the unitary matrix Q is quasiseparable of order (n_L, n_L) at most and the upper triangular matrix S is upper quasiseparable of order $n_L + n_U$ at most.*

Proof follows directly from Theorem 4.1 and Corollary 1.

5 The eigenvalue computations for Hermitian matrices

Now assume that the matrix R is Hermitian. Then the new iterant R_1 is a Hermitian matrix which is quasiseparable of the same order as the matrix R. This means that for a quasiseparable of a given order Hermitian matrix, the result of QR iteration has the same structure as the original matrix. Moreover an algorithm for computation of this structure is given.

Theorem 5.1 *Let $R = \{R_{ij}\}_{i,j=1}^{N}$ be a scalar Hermitian quasiseparable of order (n,n) matrix with lower generators p_i $(i = 2, \dots, N)$, q_j $(j = 1, \dots, N-1)$, a_k $(k = 2, \dots, N-1)$ of orders r'_k $(k = 1, \dots, N-1)$, upper generators q_i^* $(i = 1, \dots, N-1)$, p_j^* $(j = 2, \dots, N)$, a_k^* $(k = 2, \dots, N-1)$ and diagonal* *entries d_k $(k = 1, \ldots, N)$ and σ be a real number. Define the matrix R_1 by the rule*

$$\begin{cases} R - \sigma I = QS, \\ R_1 = \sigma I + SQ, \end{cases}$$

where Q is a unitary matrix and S is an upper triangular matrix.

Then R_1 is a Hermitian quasiseparable of order (n, n) at most matrix and generators and diagonal entries of this matrix are obtained using the algorithm from Theorem 4.1.

Thus for a Hermitian quasiseparable matrices R we may apply the QR iterations with shifts

$$\begin{cases} R^{(0)} = R; \\ R^{(k)} - \sigma_k I = Q^{(k)} S^{(k)}, \\ R^{(k+1)} = \sigma_k I + S^{(k)} Q^{(k)}, \quad k = 0, 1, 2, \ldots, \end{cases} \tag{5.1}$$

where $Q^{(k)}$ is a unitary matrix and $S^{(k)}$ is an upper triangular matrix, in order to compute the eigenvalues of the matrix R. Numerical examples of implementation of this algorithm with the choosing of the corresponding shifts σ_k are discussed below.

6 Complexity

Now we consider the cost of computation of the matrix R_1 in accordance with Theorem 5.1. In [6] it was shown that the the number c_1 of the flops, i.e. of arithmetic operations of the form $a \pm b$, required for the algorithm from Theorem 3.1 is estimated as follows:

$$c_1 \leq N(\vartheta(1+n, n) + \vartheta(1+n, 1+2n) + 5n^3 + 4n^2 + 2n + 1)$$

Here $\vartheta(m, r)$ means complexity of QR factorization for a matrix of sizes $m \times r$.

Next in the second stage of the algorithm from Theorem 3.2 we deal with the generators $(p_V)_i, (q_V)_i, (a_V)_i, (d_V)_i, (g_U)_i, (h_U)_i, (b_U)_i, (d_U)_i$ of sizes $1 \times \rho_{i-1}, \rho_i \times \nu_i, \rho_i \times \rho_{i-1}, 1 \times \nu_i, \nu_i \times \rho_i, \rho_{i-1} \times 1, \rho_{i-1} \times \rho_i, \nu_i \times 1$ respectively and auxiliary variables z_i, α_i, β_i which are matrices of sizes $\rho_i \times \rho_i, \rho_{i+1} \times \rho_i, \rho_i \times \rho_i$. Hence from the formulas (3.4), (3.5), (3.6) we conclude that the computation of the elements $z_i, (q_Q)_i, \alpha_i, (d_Q)_i, \beta_i$ costs, respectively $\rho_i^2 \nu_i, \rho_i \nu_i + \rho_i \rho_{i-1}, \rho_i^2 \rho_{i-1} +$

$\rho_{i+1}\rho_i^2, \nu_i + \rho_{i-1}^2 + \rho_{i-1}, \rho_i\rho_{i-1}^2 + \rho_i^2\rho_{i-1}$. Thus the total complexity of this stage

$$c_2 = \sum_{i=1}^{N}(\rho_i^2\nu_i + \rho_i\nu_i + \rho_{i-1}^2 + \rho_{i-1} + \rho_i\rho_{i-1} + 2\rho_i^2\rho_{i-1} + \rho_{i+1}\rho_i^2 + \nu_i + \rho_i\rho_{i-1}^2).$$

Similar in the second stage of the algorithm from Theorem 4.1 we deal with the generators $(p_Q)_i, (q_Q)_i, (a_Q)_i, (d_Q)_i, (g_S)_i, (h_S)_i, (b_S)_i, (d_S)_i$ of sizes $1\times\rho_{i-1}, \rho_i\times 1, \rho_i\times\rho_{i-1}, 1\times 1, 1\times\rho'_i, \rho'_{i-1}\times 1, \rho'_{i-1}\times\rho'_i, 1\times 1$ respectively and auxiliary variables z_i, α_i, β_i which are matrices of sizes $\rho'_{i-1}\times\rho_{i-1}, \rho'_{i-1}\times\rho_{i-2}, \rho'_{i-1}\times\rho_{i-1}$. Hence from the formulas (4.3), (4.4), (4.5) we conclude that the computation of the elements $z_i, p_i^{(1)}, \alpha_i, d_i^{(1)}, \beta_i$ costs, respectively $\rho'_{i-1}\rho_{i-1}, \rho_{i-1} + \rho'_i\rho'_{i-1}, (\rho'_{i-1})^2\rho'_i + \rho'_{i-1}\rho_{i-1}\rho_{i-2}, 1 + \rho'_i\rho_i + \rho_i, \rho'_i\rho'_{i-1}\rho_i + \rho'_{i-1}\rho_i\rho_{i-1}$. Thus the total complexity of this stage

$$c_3 = \sum_{i=1}^{N}(\rho'_{i-1}\rho_{i-1}+\rho_{i-1}+\rho'_i\rho'_{i-1}+(\rho'_{i-1})^2\rho'_i+\rho'_{i-1}\rho_{i-1}\rho_{i-2}+1+\rho'_i\rho_i+\rho_i+\rho'_i\rho'_{i-1}\rho_i+\rho'_{i-1}\rho_i\rho_{i-1}).$$

Now since the matrix R is quasiseparable of order (n, n) using (3.11), (3.12) and the definition of the sizes ν_k from Theorem 3.1 we have

$$\rho_i \leq n, \quad \rho'_i \leq 2n, \quad \sum_{i=1}^{N}\nu_i = N.$$

Hence it follows that the complexities c_2, c_3 satisfy the inequalities

$$c_2 \leq N(4n^3 + 3n^2 + 2n + 1), \quad c_3 \leq N(16n^3 + 8n^2 + 2n).$$

Thus the total complexity of the algorithm from Theorem 5.1 is estimated as follows:

$$c \leq N(\vartheta(1+n, n) + \vartheta(1+n, 1+2n) + 25n^3 + 15n^2 + 6n + 2).$$

Thus the algorithm of computation of the matrix R_1 has a linear $O(N)$ complexity.

7 Numerical tests

In this section we present the results of numerical tests to check the efficiency of the QR iteration method (5.1) for symmetric quasiseparable matrices based on the algorithm suggested in Theorem 4.1. Numerical tests were implemented in the system MATLAB, version 6.5.1.199709 with unit round-off error $\upsilon = 2.2204\times 10^{-16}$. In all experiments performed the input data were taken randomly using the random-function.

The QR iteration procedure was performed using the standard shift strategy (see for instance [13, p. 394]) with the use of quasiseparable structure in a similar way as in [1]. More precisely we proceed as follows. In the first two iterations we take the shift parameter σ is equal to zero. Next for the iterant $R^{(s)} = \{R_{ij}^{(s)}\}_{i,j=1}^{N}$ with generators $p_i^{(s)}$ $(i = 2, \dots, N)$, $q_j^{(s)}$ $(j = 1, \dots, N-1)$, $a_k^{(s)}$ $(k = 2, \dots, N-1)$ and the diagonal entries $d_k^{(s)}$ $(k = 1, \dots, N)$ we check the condition

$$|d_N^{(s)} - d_N^{(s-1)}| \leq 0.1|d_N^{(s)}| \tag{7.1}$$

and we set $\sigma^{(s)} = d_N^{(s)}$ if this is true and $\sigma^{(s)} = 0$ otherwise. We say that $d_N^{(s)} = R_{NN}^{(s)}$ provides a numerical approximation of an eigenvalue λ of R whenever

$$\|p_N^{(s)}\| \max\{\|a_{N-1}^{(s)}\|, \|q_{N-1}^{(s)}\|\} \leq \upsilon |d_N^{(s)}|.$$

If this condition is fulfilled, then we set $\lambda = d_N^{(s)}$ and we deflate the matrix by restarting the process with the initial matrix $R^{(s)}(1 : N-1, 1 : N-1)$, i.e. with the leading principal submatrix of $R^{(s)}$ of order $N-1$, with the choosing of the shifts in accordance with (7.1). The maximal number of iteration for every eigenvalue in our computations was limited by 35 but in all our experiments the maximal value required for an eigenvalue turns out to be equal to 24. In the end of the algorithm when we arrive to a 2×2 matrix we compute the two remaining eigenvalues using the MATLAB function *eig*.

The accuracy of the algorithm is estimated by the relations

$$\varepsilon_r = \max_i \frac{\min_j |\lambda_i - \hat{\lambda}_j|}{|\lambda_i|}, \quad \varepsilon_n = \max_i \frac{\min_j |\lambda_i - \hat{\lambda}_j|}{\|R\|},$$

where λ_i is an eigenvalue obtained via the MATLAB function *eig*, every λ_i is assumed to be exact,$\hat{\lambda}_j$, $j = 1, \dots, N$ are the approximations of eigenvalues obtained by our algorithm and $\|R\|$ is the Frobenius norm of the matrix R. In each case we present also the total number of iterations n' and the maximal number n'' of iteration required to compute a single eigenvalue.

1. The first series of experiments was performed for Hermitian quasiseparable of order two matrices. The values of elements of p, q were chosen in the range of 0 to 10, the values of the elements of a were in the range of 0 to 1 and the values of the diagonal d were taken from the range of 0 to 100. The results of computations are presented in Table 1.

Table 1. $n_L = n_U = 2$

N	$\|R\|$	ε_r	ε_n	n'	n''
20	1016	3e-14	2e-15	66	12
50	25230	4e-14	1e-14	157	14
100	5085	1e-12	1e-14	304	14
150	126810	6e-13	5e-14	498	23
200	58649	1e-12	4e-14	576	9
500	78596	1e-12	2e-14	1418	14
1000	156390	1e-12	7e-14	3012	24

2. The second series of experiments was performed for Hermitian diagonal plus semiseparable of order one matrices. The values of elements of p, q were chosen in the range of 0 to 10, the values of the elements of a were set to be equal one and the values of the diagonal d were taken from the range of 0 to 100. The results of computations are presented in Table 2.

Table 2. $n_L = n_U = 1,\ a_k = b_k = 1$

N	$\|R\|$	ε_r	ε_n	n'	n''
20	714.2	2e-14	2e-15	57	6
50	1311	2e-13	6e-15	144	10
100	3347	5e-14	9e-15	299	16
150	4469	4e-13	3e-15	438	15
200	6197	8e-13	1e-14	570	17
500	15394	6e-13	4e-14	1341	12
1000	3070	1e-12	1e-13	2781	16

Thus the results of numerical tests show that the total number of iteration to compute the eigenvalues of an $N \times N$ quasiseparable matrix is about $3N$.

References

[1] D. Bini, L. Gemignani and V. Pan, *QR*-like algorithms for generalized semiseparable matrices. *Tech. Report 1470* Department of Mathematics, University of Pisa, 2004.

[2] S. Chandrasekaran and M. Gu, Fast and stable eigendecomposition of symmetric banded plus semi-separable matrices. *Linear Algebra and Applications* 313: 107-114 (2000).

[3] S. Chandrasekaran and M. Gu, A divide-and-conquer algorithm for the

eigendecomposition of symmetric block- diagonal plus semi-separable matrices. *Numerische Mathematik* 96: 107-114 (2000).

[4] P. M. Dewilde and A. J. van der Veen, *Time-varying Systems and Computations*, Kluwer 1998.

[5] Y. Eidelman and I. Gohberg, Linear complexity inversion algorithms for a class of structured matrices. *Integral Equations and Operator Theory* 35: 28-52 (1999).

[6] Y. Eidelman and I. Gohberg, A modification of the Dewilde-van der Veen method for inversion of finite structured matrices. *Linear Algebra and Application* 343-344: 419-450 (2002).

[7] D. Fasino, N. Mastronardi and M. Van Barel, Fast and stable algorithms for reducing diagonal plus semiseparable matrices to tridiagonal and bidiagonal form. *Contemporary Mathematics* 323: 105-118 (2003).

[8] F. R. Gantmacher and M. G. Krein. Sur les matrices oscillatoires et compltement non ngatives. *Compositio Mathematica*, 4:445-476, 1937. 54, 73

[9] N. Mastronardi, S. Chandrasekaran and S. Van Huffel, Fast and stable reduction of diagonal plus semiseparable matrices to tridiagonal and bidiagonal form. *BIT* 41: 149-157 (2001).

[10] N. Mastronardi, E. Van Camp and M. Van Barel, Divide and conquer type algorithms for computing the eigendecomposition of diagonal plus semiseparable matrices. *Tech. Report 7(5/2003)*, Instituto per le Applicazioni del Calcolo "M. Picone", Consiglio Nazionale delle Ricerche, Rome, Italy, 2003.

[11] E. E. Tyrtyshnikov, Mosaic ranks for weakly semiseparable matrices. *Large-Scale Scientific Computations of Engineering and Environmental Problems II (M.Griebel, S.Margenov and P.Yalamov, eds.) Notes on numerical fluid mechanics, Vieweg*, 73: 36-41 (2000).

[12] R. Vandebril, *Semiseparable Matrices and the Symmetric Eigenvalue Problem*, Katho- lieke Universiteit Leuven, 2004.

[13] J. Stoer, R. Bulirsch, *Introduction to numerical Analysis*, Springer-Verlag, New York London, 1993.