Electronic Transactions on Numerical Analysis.
Volume 35, pp.129-147, 2009.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

ON THE FAST REDUCTION OF SYMMETRIC RATIONALLY GENERATED TOEPLITZ MATRICES TO TRIDIAGONAL FORM*

K. FREDERIX†, L. GEMIGNANI‡, AND M. VAN BAREL†

Abstract. In this paper two fast algorithms that use orthogonal similarity transformations to convert a symmetric rationally generated Toeplitz matrix to tridiagonal form are developed, as a means of finding the eigenvalues of the matrix efficiently. The reduction algorithms achieve cost efficiency by exploiting the rank structure of the input Toeplitz matrix. The proposed algorithms differ in the choice of the generator set for the rank structure of the input Toeplitz matrix.

Key words. Toeplitz matrices, eigenvalue computation, rank structures

AMS subject classifications. 65F15

1. Introduction.

The design of fast algorithms for Toeplitz matrices is a wide, active research field in structured numerical linear algebra. One of the most fruitful ideas relies upon the exploitation of the relationships between the properties of Toeplitz matrices and Laurent series, whose domain is the unit circle in the complex plane. An up-to-date survey of this beautiful mathematical theory can be found in [6]. For a given complex function $f(z) = \sum_{j=-\infty}^{+\infty} t_j z^j$ defined for $|z| = 1$ we denote $T_n = (t_{j-i})_{i,j=1}^n$ the $n \times n$ Toeplitz matrix generated by the function $f(z)$, known as the *symbol* of T_n, $n \geq 1$. The representation of a Toeplitz matrix by its symbol is a way to capture the structure which enables the initial matrix problem to be recast into a functional setting.

The knowledge of the eigenvalues and the singular values of Toeplitz matrices is of considerable interest in many applications, especially time series analysis and signal processing; see [35, 36, 37] and the references given therein. Efficient algorithms have been devised for Hermitian Toeplitz matrices generated by a Laurent polynomial or a rational function.

The methods by Trench [34, 33] and by Bini, Pan and Di Benedetto [4, 3, 5] employ the specific form of the generating function to efficiently evaluate the characteristic polynomial $p_n(z) = \det(zI - T_n)$ and/or the Newton ratio $p_n(z)/p'_n(z)$. The resulting methods are suited for the computation of a few selected eigenvalues of T_n.

Alternatively, the eigenvalue algorithms proposed in [1, 27] and [15] for banded and rationally generated symmetric Toeplitz matrices, respectively, can be used to compute the whole eigensystem of T_n. Here the approach is to find an approximation of the input Toeplitz matrix T_n in a certain algebra of matrices that are simultaneously diagonalized by a fast trigonometric transform. In the rational case the eigenproblem for T_n is thus converted to a generalized eigenproblem for the matrix pencil $H_n^{(1)} + Z_n^{(1)} - z(H_n^{(2)} + Z_n^{(2)})$, where for $i = 1, 2$, $H_n^{(i)}$ belongs to the considered matrix algebra and $Z_n^{(i)}$ is of small rank. By similarity the pencil is further transformed into the modified form $D_n^{(1)} + \widehat{Z}_n^{(1)} - z(D_n^{(2)} + \widehat{Z}_n^{(2)})$,

*Received April 15, 2008. Accepted for publication March 16, 2009. Published online June 24, 2009. Recommended by G. Ammar.

†Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium ({Katrijn.Frederix,Marc.VanBarel}@cs.kuleuven.be). The research was partially supported by the Research Council K.U.Leuven, project OT/05/40 (Large rank structured matrix computations), CoE EF/05/006 Optimization in Engineering (OPTEC), by the Fund for Scientific Research–Flanders (Belgium), G.0455.0 (RHPH: Riemann-Hilbert problems, random matrices and Padé-Hermite approximation), G.0423.05 (RAM: Rational modelling: optimal conditioning and stable algorithms), and by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Belgian Network DYSCO (Dynamical Systems, Control, and Optimization). The scientific responsibility rests with its authors.

‡Dipartimento di Matematica, Università di Pisa, Largo Bruno Pontecorvo 5, 56127 Pisa, Italy (gemignan@dm.unipi.it). This work has been supported by MIUR under project 2006017542.

ETNA
Kent State University
http://etna.math.kent.edu

where for $i = 1, 2$, $D_n^{(i)}$ is diagonal and $\operatorname{rank}(\widehat{Z}_n^{(i)}) = \operatorname{rank}(Z_n^{(i)})$.

The latter (generalized) eigenproblem can be addressed by performing a sequence of successive rank-one updates. The eigensystem of a matrix (pencil) modified by a rank-one correction is obtained by solving the associated *secular equation* [24, 7]. The caveat of this strategy is that at each step the complete eigensystem of the unperturbed matrix (pencil) is required. It is well known [25, 41] that computing the eigenvectors of a matrix can be prone to numerical instabilities and ill-conditioning problems even if the matrix is Hermitian. For this reason, the method is not recommended whenever only the eigenvalues of T_n are sought. If, otherwise, we are interested in computing both the eigenvalues and the eigenvectors of T_n then special techniques such as in [26] should be considered in the practical implementation of the updating process.

In this paper we propose a novel eigenvalue algorithm for symmetric rationally generated Toeplitz matrices based on the matrix technology for *rank-structured* matrices. The systematic study of this class of structured matrices was initiated in [19, 20, 21, 22], in the monograph [14] and in [38]. The interested reader can consult the books [39, 40] for more details concerning rank structured matrices. The approximate rank-structure of general Toeplitz matrices has been investigated in [30, 42, 31] for the purpose of finding efficient direct and iterative linear solvers.

First, the interplay between Toeplitz matrices and Laurent series is used to establish the exact rank structure of rationally generated Toeplitz matrices. Then, we develop efficient algorithms to compute the *generators* of the rank structure from the coefficients of the Laurent polynomials defining the rational symbol. Two generator sets associated with two different representations of the rank structures are specifically analyzed. Finally, we adapt the algorithms developed in [8, 18] and [12] to efficiently transform by similarity the symmetric Toeplitz matrix represented in condensed form via the generators of its rank structure into a tridiagonal form. Efficient available QR implementations can be used to compute all the eigenvalues of a Hermitian tridiagonal matrix using $O(n^2)$ flops.

The complexity of our composite eigensolvers depends on the size n of the matrix and on its rank structure. It is shown that the rank structure can be specified by $O(n \cdot g(l, m))$ parameters, where l and m denote the degrees of the numerator and the denominator of the symbol, respectively, and $g(x, y)$ is a polynomial of low degree independent of n, l and m. If, as is usual in applications, $n \gg \max\{l, m\}$ then the overall cost of our eigenvalue algorithms is $O(n^2)$. Furthermore, all the computations are carried out using unitary transformations and, therefore, the algorithms are both fast and numerically robust.

The paper is organized as follows. In Section 2, we provide a description of the rank structure of symmetric rationally generated Toeplitz matrices. In Section 3, we develop fast algorithms to compute a condensed representation for this structure and to transform by unitary similarity the input Toeplitz matrix represented via its generators into a tridiagonal form. In Section 4, an alternative tridiagonalization procedure dealing with a different generator set is presented. In Section 5, we discuss the practical implementation of our eigenvalue algorithms and report the results of numerical experiments and comparisons. Finally, our conclusions are stated in Section 6.

2. Rank structure of symmetric rationally generated Toeplitz matrices. Let

$$a(z) = a_0 + a_1 z + \ldots + a_q z^q, \quad c(z) = c_l z^{-l} + \ldots + c_1 z^{-1} + c_0 + c_1 z + \ldots + c_l z^l,$$

be two real Laurent polynomials, where $a_0, \ldots, a_q$ and $c_0, \ldots, c_l$ are real, $a_q, c_l \neq 0$, and, moreover, $a(z)$ has no zeros in $|z| \leq 1$. Then the rational function

$$t(z) = \frac{c(z)}{a(z)a(1/z)}$$

ETNA
Kent State University
http://etna.math.kent.edu

admits a Laurent expansion

$$t(z) = \sum_{j=-\infty}^{\infty} t_{|j|} z^j, \quad t_j \in \mathbb{R},$$

in an open annulus around the unit circle in the complex plane [28].

Here we investigate the rank structure of the symmetric rationally generated Toeplitz matrices

$$T_n = \begin{bmatrix} t_0 & t_1 & \dots & t_{n-1} \\ t_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_1 \\ t_{n-1} & \dots & t_1 & t_0 \end{bmatrix} \in \mathbb{R}^{n\times n},$$

for increasing n.

The first result gives a useful decomposition of $t(z)$ as considered in [16]. For the sake of simplicity, in the sequel a (Laurent) polynomial of negative degree is understood to be the zero polynomial and, similarly, a banded matrix with negative bandwidth reduces to the zero matrix.

THEOREM 2.1. *There exist a polynomial $p(z)$ of degree at most q and a symmetric Laurent polynomial $s(z)$ of degree at most $l-q$ such that*

$$c(z) = s(z)a(z)a(1/z) + p(1/z)a(z) + p(z)a(1/z), \tag{2.1}$$

which implies

$$t(z) = \frac{c(z)}{a(z)a(1/z)} = s(z) + \frac{p(1/z)}{a(1/z)} + \frac{p(z)}{a(z)}. \tag{2.2}$$

Proof. We first determine $s(z) = \sum_{i=q-l}^{l-q} s_{|i|} z^i$ by imposing that $q(z) = c(z) - s(z)a(z)a(1/z)$ has degree less than or equal to q. From $a_q, a_0 \neq 0$ it follows that $a(z)a(1/z) = \sum_{i=-q}^{q} \gamma_{|i|} z^i$ is a symmetric Laurent polynomial of degree exactly q, that is, $\gamma_q \neq 0$. The condition on the degree of $q(z) = \sum_{i=-q}^{q} \beta_{|i|} z^i$ is then equivalent to determining $s_1, \dots, s_{l-q}$ to satisfy the invertible triangular linear system

$$\begin{bmatrix} \gamma_q & & & \\ \gamma_{q-1} & \ddots & & \\ \vdots & \ddots & \ddots & \\ \gamma_{2q+1-l} & \dots & \gamma_{q-1} & \gamma_q \end{bmatrix} \begin{bmatrix} s_{l-q} \\ \vdots \\ s_1 \end{bmatrix} = \begin{bmatrix} c_l \\ \vdots \\ c_{q+1} \end{bmatrix}.$$

Now observe that the computation of $p(z) = p_0 + p_1 z + \ldots + p_q z^q$ is reduced to solving the linear system

$$\mathcal{J}\boldsymbol{p} = \boldsymbol{\beta}, \quad \boldsymbol{p}^T = [p_0, \dots, p_q], \quad \boldsymbol{\beta}^T = [\beta_0, \dots, \beta_q], \tag{2.3}$$

where $\mathcal{J} \in \mathbb{R}^{(q+1)\times(q+1)}$ is the Toeplitz-plus-Hankel matrix defined by

$$\mathcal{J} = \begin{bmatrix} a_0 & \dots & \dots & a_q \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ & & & a_0 \end{bmatrix} + \begin{bmatrix} a_0 & \dots & \dots & a_q \\ \vdots & & \iddots & \\ \vdots & \iddots & & \\ a_q & & & \end{bmatrix}. \tag{2.4}$$

Since all the zeros of $a(z)$ have modulus greater than 1 it can be shown [13] that $\mathcal{J}$ is invertible and therefore $\boldsymbol{p}$ is uniquely obtained from (2.3). $\square$

The additive decomposition (2.2) of the symbol $t(z)$ yields additive decompositions for the Toeplitz matrices T_n, $n \geq 1$, which can be used to establish their rank structures. To be precise, for any given pair of natural numbers $l \leq n$ and $m \leq n$ let us denote by $\mathcal{F}_{l,m,n} \subset \mathbb{C}^{n\times n}$ the class of $n \times n$ rank structured matrices $A = (a_{i,j}) \in \mathbb{C}^{n\times n}$ satisfying the rank constraints

$$\max_{1\leq k\leq n-1} \operatorname{rank} A(k+1\colon n, 1\colon k) \leq l, \qquad \max_{1\leq k\leq n-1} \operatorname{rank} A(1\colon k, k+1\colon n) \leq m, \tag{2.5}$$

where $B(i\colon j, k\colon l)$ is the submatrix of B with entries having row and column indices in the ranges i through j and k through l, respectively.

THEOREM 2.2. *For any $n \geq 1$, we have*

$$T_n = S_n + Q_n, \tag{2.6}$$

where S_n is a symmetric banded Toeplitz matrix with bandwidth at most $l - q$ and $Q_n \in \mathcal{F}_{q,q,n}$. Whence, it follows that $T_n \in \mathcal{F}_{m',m',n}$ with $m' = \max\{l, q\}$.

Proof. We can assume that $\deg(p(z)) < \deg(a(z))$. If, otherwise, $\deg(p(z)) = \deg(a(z)) = q$, we can consider $\hat{p}(z) = p(z) - \delta a(z)$ with δ determined so that $\deg(\hat{p}(z)) < q$. Moreover, let us suppose that the zeros $\mu_1, \ldots \mu_q$ of $a(z)$ are all distinct. Then the partial fraction decomposition of $p(z)/a(z)$ gives

$$\frac{p(z)}{a(z)} = \sum_{i=1}^{q} \frac{\rho_i}{z - \mu_i}.$$

Since $|\mu_i| > 1$ it follows that $\dfrac{\rho_i}{z - \mu_i}$ has a convergent Taylor series expansion in an open disk centered at the origin of radius greater than 1. By straightforward calculations we obtain that the rationally generated Toeplitz matrix with symbol $\dfrac{\rho_i}{z - \mu_i}$ is an upper triangular matrix belonging to $\mathcal{F}_{0,1,n}$. The proof is then completed by invoking a continuity argument to eliminate the conditions on the zeros of $a(z)$ being distinct. $\square$

It is worth noting that from the proofs of Theorems 2.1 and 2.2 it follows that the additive decomposition (2.6) is essentially unique in the sense that both S_n and Q_n are uniquely defined up to a diagonal correction which does not affect their rank structures. In the next sections the properties of this decomposition are exploited in order to design a fast and numerically robust tridiagonalization procedure for the matrix T_n.

3. Condensed representation of T_n. A basic preliminary step in the efficient reduction of symmetric rationally generated Toeplitz matrices into tridiagonal form is the computation of a condensed representation of the matrix entries, i.e., the coefficients of the associated symbol, according to the rank-structure-revealing decomposition stated in Theorem 2.2. The desired quadratic cost of the tridiagonalization scheme is achieved by working directly on this representation rather than on the input data. The rationale is that, unlike the Toeplitz-like structure, the rank structure is maintained during the process so that the amount of work does not increase significantly.

Let us assume that the rational symbol $t(z)$ is given in the form (2.2) specified by the polynomials $s(z)$, $a(z)$ and $p(z)$. Note that if we know $q = \deg(a(z))$, then these polynomials can be computed from the coefficients of the Laurent series of $t(z)$ in $O(l^2 + q^2)$ flops. The matrix S_n is a symmetric banded Toeplitz matrix of bandwidth $l - q$ and, therefore, it can be specified compactly by its matrix entries, that is, the coefficients of $s(z)$. In this and the next section we design algorithms that compute a parameterization for the rank structured matrix Q_n based on the knowledge of $a(z)$ and $p(z)$. Adding these two representations yields the input description for the matrix T_n which is modified in the tridiagonalization process.

For the sake of notational simplicity here and hereafter we restrict ourselves to the case $l \leq q$, meaning that Q_n and T_n can only differ from the elements on the diagonal, that is, $T_n = \alpha I_n + Q_n$, $n \geq 1$. The general case can be treated similarly with just some technical modifications. To represent Q_n, there are several possibilities. We can use the quasiseparable [20], the Givens-weight or the unitary-weight representation [11]. Once this representation is obtained, several algorithms can be used to solve the corresponding system of linear equations [22, 10] or to solve the eigenvalue problem [23, 9, 18, 12]. Solving the linear system can be performed in $O(q^2 n)$ flops. Solving the eigenvalue problem can be done in several ways, e.g., one can directly use the QR-algorithm on the rank structured matrix Q_n or one can transform Q_n into an orthogonally similar Hessenberg (and by symmetry tridiagonal) matrix. The reduction into a tridiagonal matrix requires $O(qn^2)$ flops.

In this section, we describe a tridiagonalization algorithm exploiting the quasiseparable representation of Q_n, whereas, in the next section an alternative approach based on the Givens-weight parametrization is presented.

3.1. The quasiseparable representation. Let T_a, T_p denote the lower triangular Toeplitz matrices (of size as appropriate in the equations) corresponding to the polynomials $a(z)$ and $p(z)$ respectively. Then we can express Q_n as follows [17]:

$$Q_n = T_a^{-1} T_p + T_p^T T_a^{-T}. \tag{3.1}$$

Note that the first term is a lower triangular Toeplitz matrix and the second term is its transpose.

A representation for the rank structure of Q_n can be easily obtained by partitioning T_a and T_p in block bidiagonal form. Suppose that $n = m \cdot q + k$, $0 \leq k < q$. The block partitioning of T_a and T_p is

$$T_a = \begin{bmatrix} \widehat{A}_0 & & & & \\ -\widehat{A}_{-1} & A_0 & & & \\ & -A_{-1} & \ddots & & \\ & & \ddots & \ddots & \\ & & & -A_{-1} & A_0 \end{bmatrix}, T_p = \begin{bmatrix} \widehat{B}_0 & & & & \\ \widehat{B}_{-1} & B_0 & & & \\ & B_{-1} & \ddots & & \\ & & \ddots & \ddots & \\ & & & B_{-1} & B_0 \end{bmatrix},$$

where $\widehat{A}_0, \widehat{B}_0 \in \mathbb{R}^{k\times k}$, $\widehat{A}_{-1}, \widehat{B}_{-1} \in \mathbb{R}^{q\times k}$, $A_j, B_j \in \mathbb{R}^{q\times q}$, $j = 0, -1$, and

$$A_0 = \begin{bmatrix} a_0 & & & \\ a_1 & a_0 & & \\ \vdots & \ddots & \ddots & \\ a_{q-1} & \dots & a_1 & a_0 \end{bmatrix}, \quad B_0 = \begin{bmatrix} p_0 & & & \\ p_1 & p_0 & & \\ \vdots & \ddots & \ddots & \\ p_{q-1} & \dots & p_1 & p_0 \end{bmatrix},$$

$$A_{-1} = -\begin{bmatrix} a_q & a_{q-1} & \dots & a_1 \\ & \ddots & \ddots & \vdots \\ & & a_q & a_{q-1} \\ & & & a_q \end{bmatrix}, \quad B_{-1} = \begin{bmatrix} p_q & p_{q-1} & \dots & p_1 \\ & \ddots & \ddots & \vdots \\ & & p_q & p_{q-1} \\ & & & p_q \end{bmatrix},$$

ETNA
Kent State University
http://etna.math.kent.edu

and

$$\widehat{A}_0 = A_0(1\colon k, 1\colon k),\ \widehat{B}_0 = B_0(1\colon k, 1\colon k),$$

$$\widehat{A}_{-1} = A_{-1}(1\colon q, q-k+1\colon q),\ \widehat{B}_{-1} = B_{-1}(1\colon q, q-k+1\colon q).$$

Let $F_a \in \mathbb{R}^{q\times q}$ be the companion matrix associated with $z^q a(z^{-1})$, i.e.,

$$F_a^T = \begin{bmatrix} -a_1/a_0 & -a_2/a_0 & \dots & -a_q/a_0 \\ 1 & 0 & \dots & 0 \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 \end{bmatrix}.$$

From Barnett's factorization [2], it follows that

$$A_{-1} \cdot A_0^{-1} = F_a^q.$$

Since the spectral radius of F_a^q is less than 1, the power sequence of the matrix tends to zero. Let $\Delta = \widehat{A}_{-1}\widehat{A}_0^{-1} \in \mathbb{R}^{q\times k}$. The inverse of T_a is the block matrix given by

$$T_a^{-1} = \left[\begin{array}{c|ccccc} \widehat{A}_0^{-1} & & & & \\ \hline A_0^{-1}\Delta & A_0^{-1} & & & \\ A_0^{-1}F_a^q\Delta & A_0^{-1}F_a^q & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ A_0^{-1}F_a^{(m-1)q}\Delta & A_0^{-1}F_a^{(m-1)q} & \dots & A_0^{-1}F_a^q & A_0^{-1} \end{array}\right].$$

Therefore, by using (3.1) we arrive at the following block condensed representation of Q_n.

THEOREM 3.1. *The symmetric Toeplitz matrix Q_n defined by (3.1) can be partitioned in a block form $Q_n = (Q_{i,j}^{(n)})_{i,j=1}^{m+1}$, where $Q_{i,j}^{(n)} \in \mathbb{R}^{n_i\times n_j}$, $n_1 = k$, $n_2 = \dots_m = q$, $Q_{i,j}^{(n)} = Q_{j-i}$ for $j \geq i \geq 2$, and*

$$Q_{i,j}^{(n)} = \begin{cases} A_0^{-1}\cdot F_a^{q(i-2)}\cdot\Gamma_0^{(n)}, & \text{if } i\geq 2,\ j=1; \\ A_0^{-1}\cdot F_a^{q(i-j-1)}\cdot\Gamma_1, & \text{if } i-j\geq 1,\ j\geq 2, \end{cases}$$

where

$$\Gamma_0^{(n)} = \Delta\widehat{B}_0 + \widehat{B}_{-1}, \quad \Gamma_1 = F_a^q B_0 + B_{-1}.$$

Representations of this form for rank structured matrices have been introduced in [14, 20] in the framework of *quasiseparable* matrices and matrices with small *Hankel rank*. In order to merge the rank structures of Q_n and S_n, we find a suitable decomposition of Q_n by performing a step of block Neville elimination. Let B_n be the block lower bidiagonal matrix partitioned commensurable with Q_n and defined by

$$B_n = \left[\begin{array}{c|cccc} I_k & & & \\ \hline & I_q & & \\ & -\Sigma & \ddots & \\ & & \ddots & \ddots & \\ & & & -\Sigma & I_q \end{array}\right], \quad \Sigma = A_0^{-1}F_a^q A_0.$$

ETNA
Kent State University
http://etna.math.kent.edu

Then we have the following theorem.

THEOREM 3.2. *The matrix $P_n = B_n \cdot Q_n \cdot B_n^T$ is a symmetric block tridiagonal matrix with subdiagonal blocks*

$$P_{2,1}^{(n)} = Q_{2,1}^{(n)}, \quad P_{i+1,i}^{(n)} = P_1^T = Q_1^T - \Sigma Q_0,\ 2 \le i \le m,$$

and diagonal blocks

$$P_{1,1}^{(n)} = Q_{1,1}^{(n)}, \quad P_{2,2}^{(n)} = Q_{2,2}^{(n)} = Q_0,$$

and

$$P_{i,i}^{(n)} = P_0 = Q_0 + \Sigma Q_0 \Sigma^T - \Sigma Q_1 - Q_1^T \Sigma^T, \quad 3 \le i \le m+1.$$

From this theorem we conclude that

$$T_n = B_n^{-1} \cdot (P_n + \alpha B_n \cdot B_n^T) \cdot B_n^{-T} = B_n^{-1} \cdot Z_n \cdot B_n^{-T}, \tag{3.2}$$

where the "middle" factor

$$Z_n = P_n + \alpha B_n \cdot B_n^T$$

is a banded matrix with bandwidth $2q-1$ at most. In the next subsection we exploit this representation of T_n for the design of an efficient tridiagonalization procedure. Note that when $l > q$ the "middle" factor $Z_n = P_n + B_n S_n B_n^T$ is a banded matrix with bandwidth $q + l - 1$ at most.

3.2. Tridiagonal reduction algorithm. In this section we describe a fast block algorithm for reducing T_n into tridiagonal form by unitary transformations. In principle the reduction may be carried out using the scalar algorithm given in [18] for rank structured matrices represented in quasiseparable form. The efficiency could be further improved by adjusting the algorithm to work directly with block rather than scalar quasiseparable representations, similarly to the approach followed in [21] for the QR factorization of rank structured matrices. Although the generalization is possible, the form (3.2) of T_n suggests the use of a different block reduction scheme related to the scalar technique proposed in [8].

The building blocks of the tridiagonalization procedure are the QR factorization of small matrices of size $O(q)$ and standard bulge-chasing schemes for banded reduction [32]. Let $B_n^{-1} = B_n^{(0)}$ and $Z_n = Z_n^{(0)}$. Moreover, let $U^{(1)} \in \mathbb{R}^{2q \times 2q}$ be an orthogonal matrix determined to satisfy

$$U^{(1)^T} = \begin{bmatrix} U_{1,1}^{(1)} & U_{1,2}^{(1)} \\ U_{2,1}^{(1)} & U_{2,2}^{(1)} \end{bmatrix}, \quad U^{(1)^T} \begin{bmatrix} I_q \\ \Sigma \end{bmatrix} = \begin{bmatrix} R^{(1)} \\ 0 \end{bmatrix},$$

where $U_{1,1}^{(1)}, U_{2,2}^{(1)} \in \mathbb{R}^{q \times q}$ and $R^{(1)} \in \mathbb{R}^{q \times q}$ is upper triangular. It is immediately seen that the block Givens-like matrix

$$\mathcal{G}^{(1)} = I_{(m-2)q+k} \oplus U^{(1)^T}$$

is such that

$$\mathcal{G}^{(1)} \cdot B_n^{(0)} = \left[\begin{array}{c|cccccc|cc} I_k & & & & & & & & \\ \hline & & & & & & & & \\ & \multicolumn{6}{c|}{B_n^{(0)}(k+1\colon (m-2)q, k+1\colon (m-2)q)} & & \\ & & & & & & & & \\ \hline & \Sigma^{m-2} R^{(1)} & \dots & \dots\dots & \dots & & \Sigma R^{(1)} & R^{(1)} & U_{1,2}^{(1)} \\ & 0 & \dots & \dots\dots & \dots & & 0 & 0 & U_{2,2}^{(1)} \end{array}\right].$$

ETNA
Kent State University
http://etna.math.kent.edu

The matrix on the right-hand side can be rewritten as

$$\left[\begin{array}{c|cccc|c} I_k & & & & & \\ \hline & \multicolumn{4}{c|}{B_n^{(0)}(k+1\colon (m-2)q, k+1\colon (m-2)q)} & \\ \cline{2-5} & \Sigma^{m-2}R^{(1)} & \dots & \dots\dots \quad \dots & \Sigma R^{(1)} & I_{2q} \\ & 0 & \dots & \dots\dots \quad \dots & 0 & \end{array}\right] \cdot (I_{(m-2)q+k} \oplus \begin{bmatrix} R^{(1)} & U_{1,2}^{(1)} \\ 0 & U_{2,2}^{(1)} \end{bmatrix}).$$

Set

$$B_n^{(1)} = \left[\begin{array}{c|cccc|c} I_k & & & & & \\ \hline & \multicolumn{4}{c|}{B_n^{(0)}(k+1\colon (m-2)q, k+1\colon (m-2)q)} & \\ \cline{2-5} & \Sigma^{m-2}R^{(1)} & \dots & \dots\dots \quad \dots & \Sigma R^{(1)} & I_{2q} \\ & 0 & \dots & \dots\dots \quad \dots & 0 & \end{array}\right],$$

and

$$Z_n^{(1)} = (I_{(m-2)q+k} \oplus \begin{bmatrix} R^{(1)} & U_{1,2}^{(1)} \\ 0 & U_{2,2}^{(1)} \end{bmatrix}) \cdot Z_n^{(0)} \cdot (I_{(m-2)q+k} \oplus \begin{bmatrix} R^{(1)} & U_{1,2}^{(1)} \\ 0 & U_{2,2}^{(1)} \end{bmatrix})^T.$$

It is found that $Z_n^{(1)}$ is still block tridiagonal.

Now let $U^{(2)} \in \mathbb{R}^{2q\times 2q}$ be the orthogonal matrix determined to satisfy

$$U^{(2)^T} = \begin{bmatrix} U_{1,1}^{(2)} & U_{1,2}^{(2)} \\ U_{2,1}^{(2)} & U_{2,2}^{(2)} \end{bmatrix}, \quad U^{(2)^T} \begin{bmatrix} I_q \\ R^{(1)}\Sigma \end{bmatrix} = \begin{bmatrix} R^{(2)} \\ 0 \end{bmatrix},$$

where $R^{(2)} \in \mathbb{R}^{q\times q}$ is upper triangular. Let us define the block Givens-like matrix $\mathcal{G}^{(2)}$ by

$$\mathcal{G}^{(2)} = I_{(m-3)q+k} \oplus U^{(2)^T} \oplus I_q.$$

Observe that

$$\mathcal{G}^{(2)} \cdot B_n^{(1)} = \left[\begin{array}{c|cccc|cc|c} I_k & & & & & & & \\ \hline & \multicolumn{4}{c|}{B_n^{(0)}(k+1\colon (m-3)q, k+1\colon (m-3)q)} & & & \\ \cline{2-8} & \Sigma^{m-3}R^{(2)} & \dots & \dots\dots \quad \dots & \Sigma R^{(2)} & R^{(2)} & U_{1,2}^{(2)} & \\ & 0 & \dots & \dots\dots \quad \dots & 0 & 0 & U_{2,2}^{(1)} & \\ \cline{2-8} & & & & & & & I_q \end{array}\right].$$

Again we can write

$$\mathcal{G}^{(2)} \cdot B_n^{(1)} = B_n^{(2)} \cdot (I_{(m-3)q+k} \oplus \begin{bmatrix} R^{(2)} & U_{1,2}^{(2)} \\ 0 & U_{2,2}^{(2)} \end{bmatrix} \oplus I_q),$$

ETNA
Kent State University
http://etna.math.kent.edu

where

$$B_n^{(2)} = \left[\begin{array}{c|cccccc|c} I_k & & & & & & & \\ \hline & & & & & & & \\ & & & B_n^{(0)}(k+1\colon (m-3)q, k+1\colon (m-3)q) & & & & \\ \cline{2-7} & \Sigma^{m-3}R^{(2)} & \dots & \dots\dots & \dots & \Sigma R^{(2)} & & \\ & 0 & \dots & \dots\dots & \dots & 0 & & I_{3q} \\ & \vdots & \dots & \dots\dots & \dots & \vdots & & \end{array}\right].$$

Set

$$Z_n^{(2)} = (I_{(m-3)q+k} \oplus \begin{bmatrix} R^{(2)} & U_{1,2}^{(2)} \\ 0 & U_{2,2}^{(2)} \end{bmatrix} \oplus I_q) \cdot Z_n^{(1)} \cdot (I_{(m-3)q+k} \oplus \begin{bmatrix} R^{(2)} & U_{1,2}^{(2)} \\ 0 & U_{2,2}^{(2)} \end{bmatrix} \oplus I_q)^T.$$

As a result of these matrix multiplications the block tridiagonal structure of $Z_n^{(1)}$ is destroyed. Specifically, we have that

$$Z_n^{(2)}((m-3)q+k\colon n, (m-3)q+k\colon n) = \begin{bmatrix} Z_{m-1,m-1}^{(2)} & Z_{m,m-1}^{(2)}{}^T & Z_{m+1,m-1}^{(2)}{}^T \\ Z_{m,m-1}^{(2)} & Z_{m,m}^{(2)} & Z_{m+1,m}^{(2)}{}^T \\ Z_{m+1,m-1}^{(2)} & Z_{m+1,m}^{(2)} & Z_{m+1,m+1}^{(2)} \end{bmatrix},$$

that is, a bulge in position $(m+1, m-1)$ and its symmetric analogue in position $(m-1, m+1)$ appear. To chase away this bulge we can determine an orthogonal matrix $W^{(1)} \in \mathbb{R}^{2q \times 2q}$ such that the matrix

$$W^{(1)T} \begin{bmatrix} Z_{m,m-1}^{(2)} & Z_{m,m}^{(2)} \\ Z_{m+1,m-1}^{(2)} & Z_{m+1,m}^{(2)} \end{bmatrix}$$

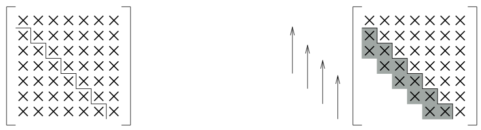
is upper triangular. Then the transformation

$$Z_n^{(2)} \leftarrow (I_{(m-2)q+k} \oplus W^{(1)}{}^T) \cdot Z_n^{(2)} \cdot (I_{(m-2)q+k} \oplus W^{(1)})$$

is used to restore the block tridiagonal structure of $Z_n^{(2)}$. It is worth noting that $B_n^{(2)}$ and $(I_{(m-2)q+k} \oplus W^{(1)})$ commute so that the process can continue in a similar fashion. The overall complexity is $O(m^2q^3) = O(n^2q)$ flops.

4. An alternative approach. In this section, an alternative method for reducing Q_n into tridiagonal form by unitary transformations is described. The proposed approach relies upon the construction of a Givens-weight representation for the rank structured matrix Q_n based on the knowledge of $a(z)$ and $p(z)$; see equation (3.1). In the following subsections, the algorithm that computes a Givens-weight representation for the rank structured matrix Q_n based on the knowledge of $a(z)$ and $p(z)$, and the algorithm to bring the matrix into Hessenberg form is explained.

4.1. Givens-weight representation. A rank structured matrix can be represented by a Givens-weight representation. It is a compact internal representation which consists of a sequence of Givens arrows which have width r (this means that the Givens arrow consists of r Givens transformations, with r the rank of the structure blocks), and a weight matrix containing compressed information about the elements in the rank structure. The weights are stored during this process of determining the sequence of Givens arrows.



(a) (b)

FIG. 4.1. *Example of a Givens-weight representation: (a) the rank structure, (b) the corresponding Givens-weight representation.*

Figure 4.1(a) shows an example of the kind of rank structured matrix which is considered in this paper. Figure 4.1(b) shows the corresponding Givens-weight representation of the rank structured matrix. At the left the arrows denote the Givens arrows (consisting of r Givens transformations, in this case $r = 2$), and the elements in gray denote the weights. The representation is internal. Therefore elements outside the rank structure are not touched. For a more detailed description about the computation of such a compact representation the interested reader is referred to [11].

Computing a Givens-weight representation for Q_n consists of finding a sequence of Givens arrows whose product is the orthogonal matrix Q such that

$$Q^T Q_n = R_q,$$

with R_q a lower banded matrix with q subdiagonals. It is important to see now that applying an orthogonal transformation Q^T to the rows of Q_n is the same as applying Q to the columns of T_a in the first term and to the columns of T_p in the second term in (3.1), i.e.,

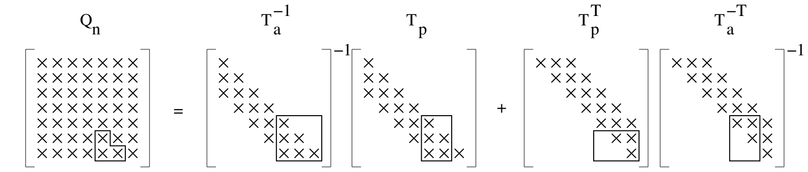
$$Q^T Q_n = (T_a Q)^{-1} T_p + (T_p Q)^T T_a^{-T}. \tag{4.1}$$

The matrix Q is the product of a sequence of Givens arrows which consist of q Givens transformations where each Givens arrow works from right to left on the columns of the matrix and makes a subdiagonal of T_a zero. It can be shown that $T_a Q$ is a nonsingular upper triangular matrix (having q nonzero superdiagonals) while $T_p Q$ is a banded matrix having q superdiagonals. Therefore $(T_a Q)^{-1} T_p$ has q subdiagonals and $(T_p Q)^T T_a^{-T}$ has q subdiagonals. The sequence of Givens arrows consisting of q Givens transformations is the Givens-part of the Givens-weight representation.

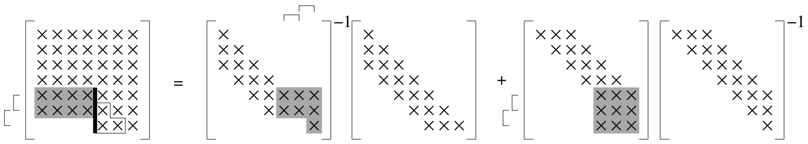
The Givens transformations and the weights are determined in the same order as when computing a Givens-weight representation, meaning going from the bottom to the top of the structure. Instead of working on the matrix Q_n, we will work on the matrix T_a by making it upper triangular to determine the weights. The algorithm is explained for a 7×7 matrix with $q = 2$. This is shown in Figure 4.2.

In fact, the algorithm only requires the information of the first term $T_a^{-1} T_p$ because the second term describes the upper triangular part of Q_n. But for completeness the result of the actions of the algorithm on the matrix Q_n and the second term $T_p^T T_a^{-T}$ are also shown. The bold box in Q_n denotes the weights which we want to compute or which already have been computed. In the other matrices (the sum), it denotes the elements we have to compute to obtain the weights. Note that T_a, T_a^T is represented as a matrix in the figures and not T_a^{-1}, T_a^{-T}, respectively.

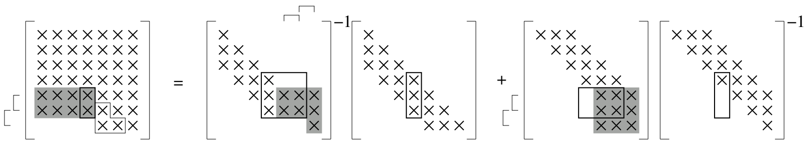
Before we can start to make T_a upper triangular, the weights of the bottom q blocks are computed, denoted in the bold box in Q_n in Figure 4.2(a). These elements lie inside the rank structure, but no Givens transformations will act on them, so the real elements will be stored. To compute these elements, only information of the elements in the two bold boxes of the



(a) Computation of the weights of the bottom q blocks.



(b) Making the bottom row of T_a upper triangular.



(c) Computation of the weight of the $q+1$ structure block.

FIG. 4.2. *Computation of the weights.*

term $T_a^{-1}T_p$ is required; see Figure 4.2(a). The product of the bold boxes in the second term will give an upper triangular matrix, so this cannot be of any influence on the elements which we want to compute.

To compute these weights, the elements of a submatrix of size $(q+1)\times(q+1)$ of T_a^{-1} are required. Let us represent the whole lower block triangular matrix T_a and its inverse by

$$T_a = \left[\begin{array}{cc} A & 0 \\ B & C \end{array}\right],\; T_a^{-1} = \left[\begin{array}{cc} A^{-1} & 0 \\ -C^{-1}BA^{-1} & C^{-1} \end{array}\right],$$

with $A \in \mathbb{C}^{(n-q-1)\times(n-q-1)}$, $B \in \mathbb{C}^{(q+1)\times(n-q-1)}$ and $C \in \mathbb{C}^{(q+1)\times(q+1)}$. Matrix C^{-1} is the submatrix (denoted in the bold box in the first term of Figure 4.2(a)) we need to compute the weights. Instead of inverting the whole matrix T_a, only the inversion of a small submatrix is necessary. The weights of the q bottom structure blocks are obtained by multiplying the inverse of C with the corresponding columns of T_p. This is shown in Figure 4.2(a). This step is a preparation step because no Givens transformations were computed.

Now the actual construction of the Givens-weight representation is explained. In general, the matrix T_a will be successively made upper triangular by applying Givens transformations and the corresponding weights will be computed. The process starts with creating zeros in a specific row of T_a (this row corresponds to the row in the matrix Q_n, where we want to create zeros) by applying q Givens transformations to the columns of this matrix. To be complete, the transposed Givens transformations have to be applied to the matrix Q_n and also to the second term; this is shown in Figure 4.2(b) (gray elements denote compressed elements). It is considered that the transposed Givens transformation on Q_n is only applied to the columns inside the rank structure. This limited number of columns is called the action radius of the Givens transformation. The action radius is denoted with a bold line in Figure 4.2(b).

Now it is possible to compute the weight of the structure block. This time the second term also has no influence on the weight. To compute the weight, the inverse of a submatrix of T_aQ_1 (Q_1 is the product of the q Givens transformations already applied to the columns

ETNA
Kent State University
http://etna.math.kent.edu

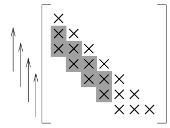


FIG. 4.3. *Givens-weight representation for Q_n.*

of the matrix) of size $(q+1)\times(q+1)$, C, is needed. T_aQ_1 is upper block triangular and we represent the matrix and its inverse as follows:

$$T_aQ_1 = \begin{bmatrix} A & 0 & 0 \\ B & C & D \\ 0 & 0 & E \end{bmatrix}, \; (T_aQ_1)^{-1} = \begin{bmatrix} A^{-1} & 0 & 0 \\ -C^{-1}BA^{-1} & C^{-1} & -C^{-1}DE^{-1} \\ 0 & 0 & E^{-1} \end{bmatrix},$$

with $A \in \mathbb{C}^{(n-q-2)\times(n-q-2)}$, $B \in \mathbb{C}^{(q+1)\times(n-q-2)}$, $C \in \mathbb{C}^{(q+1)\times(q+1)}$, $D \in \mathbb{C}^{(q+1)\times 1}$ and $E \in \mathbb{C}$. Only the inverse of the small matrix C, of size $(q+1)\times(q+1)$, is required to compute the weight. When this inverse is computed, it can be multiplied with the corresponding column of T_p to obtain the weight of the structure block; see Figure 4.2(c).

This process of making a row of T_a upper triangular and then computing the weight by inverting a small submatrix of size $q+1\times q+1$ and multiplying it to the corresponding column of T_p is continued until all the weights of the blocks in the rank structure are computed. After each step, the bold box in $T_a^{-1}T_p$ will move up along the diagonal by one element. The result of the algorithm is shown in Figure 4.3. The q Givens transformations which belong to one weight are represented by a Givens arrow of width q.

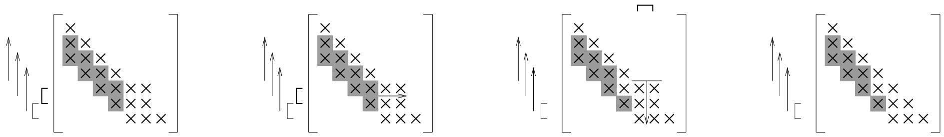
4.2. Tridiagonal reduction algorithm. The next step is to reduce the rank structured matrix with the corresponding Givens-weight representation into a Hessenberg (and by symmetry tridiagonal) matrix. To do this the method to transform a given matrix with a Givens-weight representation into a Hessenberg matrix discussed in [12] is simplified. In [12], the given Givens-weight representation is transformed into a zero-creating Givens-weight representation and then the matrix is brought in Hessenberg form by peeling off the tails of the Givens transformations meanwhile making the structure blocks one-by-one upper triangular, or in other words, bringing the columns in Hessenberg form.

In this paper, the transformation to a zero-creating Givens-weight representation is omitted and the process is not going to peel off the tails of the Givens transformations. The process is split into two parts: in the first part the precomputed Givens arrows are applied outside the rank structure and in the second part the matrix is brought into Hessenberg form.

During the first part, the precomputed Givens transformations are applied in the same order as when constructing the Givens-weight representation (§4.1) to the elements outside the rank structure. This means that the Givens transformations are applied successively outside their action radii. During the second part, each column is brought into Hessenberg form by applying a Householder transformation. Also the symmetry of the matrix will be exploited.

The initial situation of the algorithm is shown in Figure 4.3 (this is the end situation of the construction of the Givens-weight representation). For each weight, there are q precomputed Givens transformations. These are combined in a Givens arrow of width q, and each Givens arrow has a specific action radius. The algorithm applies the computed Givens arrows of the Givens-weight representation successively to the elements outside the rank structure (or in other words, outside the action radius) in the same order as they were computed. In order to preserve the eigenvalue spectrum, the transposes of the Givens transformations are applied to the columns.

ETNA
Kent State University
http://etna.math.kent.edu



(a) (b) (c) (d)

FIG. 4.4. *Exploiting symmetry. (a) Compute the required superdiagonal element, and fill it in into the weight matrix, (b) Apply the current Givens transformation to the rows, (c) Apply the transposed Givens transformation to the columns, (d) The current superdiagonal element has no role anymore, and therefore it is removed from the weight matrix.*

In Figure 4.4, the exploitation of the symmetry is explained. Only the diagonal and the q subdiagonals are considered, since the rest is known by symmetry. The first Givens arrow is decomposed into its q Givens transformations. When we want to apply the current Givens transformation (denoted in bold) to the rows, the corresponding superdiagonal element has to be computed (by symmetry) and has to be filled in into the weight matrix. This is shown in Figure 4.4(a).

Then the Givens transformation can be applied outside the action radius until the column of the added superdiagonal element; see Figure 4.4(b). Notice that after this the top element of the corresponding weight element is turned from gray to white; see Figure 4.4(c). This weight element is "completely released", meaning that no more Givens transformations act on it (the other Givens transformations have smaller action radii). To complete the similarity transformation, the transposed Givens transformation has to be applied to the columns; see Figure 4.4(c). Now the superdiagonal element has no role anymore, therefore it is removed from the weight matrix; see Figure 4.4(d).

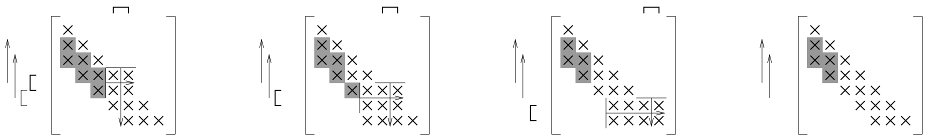
The same principle as explained in Figure 4.4 is used during the whole algorithm. So, the same principle is done for the second Givens transformation of the first Givens arrow. The result is shown in Figure 4.5(a). Notice that after the application of the Givens arrow, the corresponding weight is "completely released".

Starting from Figure 4.5(a), the flow of the algorithm is explained. Apply the q Givens transformations of the current Givens arrow outside their action radius, as explained in Figure 4.4 (this is shown in Figure 4.5(a)- 4.5(b)). Notice that when this is done, the matrix has no q subdiagonals anymore, and some fill-in elements appear in the matrix. This is shown in Figure 4.5(b). The next step is to remove these fill-in elements (in Figure 4.5(b) there is only one fill-in element located at position $(7, 4)$) by applying Givens transformations to create again a matrix with q subdiagonals. Because of similarity reasons, the transposed Givens transformation is also applied to the columns. This process is shown in Figure 4.5(c)-4.5(d).

This process of applying the Givens transformations outside the rank structure and removing the fill-in elements is continued until all the Givens arrows are applied outside their action radius. At the end, a matrix $R = QQ_nQ^H$ with q subdiagonals is obtained. Then the matrix $R + \alpha I$ has to be transformed into a Hessenberg (or by symmetry tridiagonal) matrix.

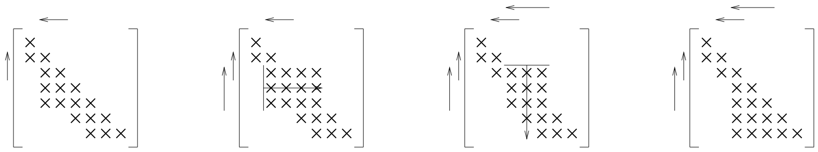
Note that when $l > q$ the matrix S_n has to be updated under the action of the Givens transformations of the Givens-weight representation of Q_n: $S = QS_nQ^H$. The matrix S is a matrix with l subdiagonals. The sum of the two matrices R and S results in a matrix with l subdiagonals which has to be transformed into a Hessenberg matrix.

To bring the matrix into Hessenberg form the columns are brought one-by-one into Hessenberg form, this time starting at the top of the structure. Again the symmetry is exploited. Figure 4.6(a) gives the matrix when the first column has already been brought into Hessenberg form. This is done by applying a Householder transformation. Notice that there is a



(a) (b) (c) (d)

FIG. 4.5. *Process to apply the Givens transformation outside the rank structure. (a) Apply the current Givens transformation to the rows and the transposed Givens transformation to the columns, remove the current superdiagonal element, (b) Apply the current Givens transformation to the rows and the transposed Givens transformation to the columns, remove the current superdiagonal element, (c) Remove the fill-in element by applying a Givens transformation to the rows and also the transposed Givens transformation to the columns, (d) The matrix has again q subdiagonals. The next Givens transformation can be applied.*



(a) (b) (c) (d)

FIG. 4.6. *Process to bring the matrix into Hessenberg form. (a) First column has been brought in Hessenberg form, (b) Apply Givens transformation to make block upper triangular, (c) Apply the transposed Givens transformation to the columns, (d) Remove superdiagonal element, notice that there has been some fill in.*

fill-in element in position $(5, 2)$.

Now the second Householder transformation has to be applied for bringing the second column into Hessenberg form (second structure block has to become upper triangular). Before this can be done some superdiagonal elements have to be added; see Figure 4.6(b). To complete the similarity transformation the Hermitian transposed operation has to be applied to the columns; see Figure 4.6(c). After this there will be fill-in elements in columns 3 and 4; see Figure 4.6(d). These will be removed when the next column is brought into Hessenberg form by applying another Householder transformation.

This process is continued until the Hessenberg form is obtained. Now efficient algorithms can be used to compute the eigenvalues of the matrix.

5. Numerical results. To check the accuracy and the numerical stability of the proposed fast tridiagonalization algorithms, we have performed several numerical experiments. For the sake of comparison the algorithm of Section 3, named *alg_1*, exploiting the quasiseparable representation of the input matrix entries and the algorithm of Section 4, referred to as *alg_2*, dealing with the Givens-weight representation of these entries have been implemented in MATLAB[1]. To test the proposed algorithms, first three typical numerical examples taken from [35] are tested and then more specific test problems are considered. The first three examples are the following:

1. The Toeplitz matrix (Kac, Murdock and Szegö [29]) considered is:

$$T_n = (0.5^{|i-j|})_{i,j=1}^n.$$

The corresponding rational function is

$$t(z) = \frac{0.75}{(1-0.5z)(1-0.5z^{-1})}.$$

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

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 5.1
Numerical errors generated by alg_1 *for example* 1, 2, 3.

n	Example 1	Example 2	Example 3
10	1.0×10^{-15}	6.4×10^{-16}	1.6×10^{-15}
50	2.0×10^{-15}	1.2×10^{-15}	3.2×10^{-15}
100	4.1×10^{-15}	1.7×10^{-15}	3.3×10^{-15}
500	1.4×10^{-14}	3.5×10^{-15}	1.0×10^{-14}
1000	2.3×10^{-14}	5.6×10^{-15}	1.6×10^{-14}

TABLE 5.2
Numerical errors generated by alg_2 *for example* 1, 2, 3.

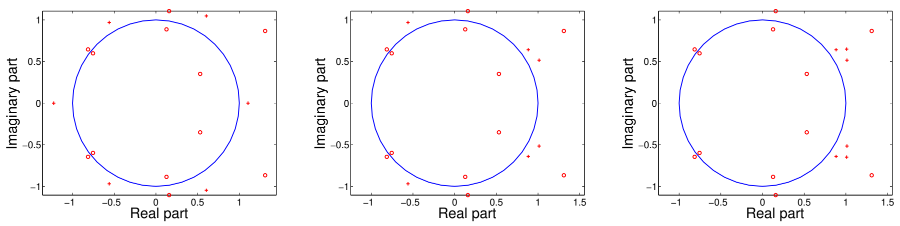
n	Example 1	Example 2	Example 3
10	5.2×10^{-16}	6.6×10^{-16}	1.3×10^{-15}
50	1.1×10^{-15}	1.3×10^{-15}	2.6×10^{-15}
100	1.4×10^{-15}	1.2×10^{-15}	4.1×10^{-15}
500	1.7×10^{-15}	4.1×10^{-15}	8.2×10^{-15}
1000	1.6×10^{-15}	4.0×10^{-15}	1.8×10^{-15}

2. The rational function is
$$t(z) = \frac{z^{-2} - 3.5z^{-1} + 1.5 - 3.5z + z^2}{a(z)a(z^{-1})},$$
where $a(z) = (1 - 0.1z)(1 - 0.2z)$.
3. The rational function is
$$t(z) = \frac{z^{-3} - z^{-2} + 2z^{-1} + 1 + 2z - z^2 + z^3}{a(z)a(z^{-1})},$$
where $a(z) = 1 - 0.4z - 0.47z^2 + 0.21z^3$.

The computed eigenvalues are compared to the exact eigenvalues of the matrix T_n, which are computed with the function `eig` in MATLAB. The results of the numerical experiments of these three examples are shown in Table 5.1 and Table 5.2 for algorithm *alg_1* and *alg_2*, respectively. Specifically, the tables contain the relative errors (in norm) between the computed and exact eigenvalues for the three examples and different matrix sizes. The accuracy of the two algorithms is comparable, the computed eigenvalues are very accurate in all the cases, and the error increases slightly when the matrix size increases.

The previous three examples are simple examples because the values for q are small $q = 1, 2, 3$, ($l = 0, 2, 3$). Therefore other specific problems will be tested. For a specific value of q, we will distinguish three different cases for the zeros of the polynomial $a(z)$ (these are the poles of $t(z)$). The polynomial $c(z)$ does not vary in the three cases and its degree equals the degree of polynomial $a(z)$ ($l = q$). The zeros of the polynomial are chosen outside but close to the unit circle in three different ways. In case 1, the argument of the poles are normally distributed around the unit circle; in case 2, some zeros of $a(z)$ are clustered together but there are still zeros at the left of the unit circle; and in case 3 all the zeros are located at one side of the unit circle. Figure 5.1 shows the localization of the zeros of $a(z)$ and $c(z)$ for $q = 6$.

The main goal of these numerical experiments is the investigation of the behaviour of the fast algorithms under less favourable conditions. In particular, as an effect of the location of the poles it is seen that the coefficients of the Laurent expansion of the rational function $1/(a(z)a(z^{-1}))$ vary much in magnitude. This implies that the inverse of the Jury matrix $\mathcal{J}$ in (2.4) can also have a large norm thus yielding a large absolute error in the computed



(a) Case 1 (b) Case 2 (c) Case 3

FIG. 5.1. *Localization of the zeros of $a(z)$ and $c(z)$ around the unit circle for $q = 6$. A plus sign denotes a zero of $a(z)$, and a circle denotes a zero of $c(z)$.*

solution $\hat{\boldsymbol{p}}$ of the linear system (2.3). Suppose that $\hat{\boldsymbol{p}} - \boldsymbol{p} = \boldsymbol{\delta}$. Then from

$$\frac{\hat{p}(1/z)}{a(1/z)} + \frac{\hat{p}(z)}{a(z)} = \frac{p(1/z)}{a(1/z)} + \frac{p(z)}{a(z)} + \frac{\delta(1/z)a(z) + \delta(z)a(1/z)}{a(z)a(1/z)},$$

it follows that $\|\Delta\|$, $\Delta = \hat{T}_n - T_n$, can be large, too. The matrix $\hat{T}_n$ denotes the Toeplitz matrix generated by the perturbed symbol $\frac{\hat{p}(1/z)}{a(1/z)} + \frac{\hat{p}(z)}{a(z)}$. Specifically, a rough qualitative estimation says that the perturbation error should be of order $\kappa(\mathcal{J})\|T_n\|$, where $\kappa(\mathcal{J})$ denotes the condition number of $\mathcal{J}$, which gives a relative error of order $\kappa(\mathcal{J})$.

Table 5.3-5.8 shows the results for three different values of the degree of polynomial $a(z)$ ($q = 6, 10, 20$). Each table contains the results for the three different cases described above and for different matrix sizes. The condition number of the matrix $\mathcal{J}$ is also reported in the bottom row of the tables.

The experimental results displayed in the tables are in good accordance with the theoretical expectations. Both algorithms are numerically robust and the condition number of the matrix $\mathcal{J}$ gives a good indication of the loss in accuracy in the computed eigenvalues. It can also be seen that the accuracy slightly increases when the matrix size increases.

TABLE 5.3
Numerical errors generated by alg_1 *for Example $q = 6$ in the three different cases.*

n	Case 1	Case 2	Case 3
100	2.9×10^{-15}	2.5×10^{-12}	6.8×10^{-9}
500	4.7×10^{-15}	2.6×10^{-12}	7.3×10^{-9}
1000	6.8×10^{-15}	2.6×10^{-12}	7.5×10^{-9}
$\kappa(\mathcal{J})$	5.8×10^{0}	2.0×10^{3}	4.4×10^{6}

TABLE 5.4
Numerical errors generated by alg_2 *for Example $q = 6$ in the three different cases.*

n	Case 1	Case 2	Case 3
100	1.3×10^{-15}	7.8×10^{-13}	2.0×10^{-9}
500	3.1×10^{-15}	8.6×10^{-13}	2.9×10^{-9}
1000	3.1×10^{-15}	1.0×10^{-12}	3.3×10^{-9}
$\kappa(\mathcal{J})$	5.8×10^{0}	2.0×10^{3}	4.4×10^{6}

6. Conclusion. We introduced two novel $O(n^2)$ fast algorithms to reduce an $n \times n$ symmetric rationally generated Toeplitz matrix into tridiagonal form by unitary transformations. Both algorithms rely upon the exploitation of the rank structures of the Toeplitz matrix that

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 5.5
Numerical errors generated by alg_1 *for Example* $q = 10$ *in the three different cases.*

n	Case 1	Case 2	Case 3
100	1.7×10^{-15}	2.2×10^{-13}	1.1×10^{-5}
500	3.1×10^{-15}	3.4×10^{-13}	1.2×10^{-5}
1000	5.1×10^{-15}	4.0×10^{-13}	1.3×10^{-5}
$\kappa(\mathcal{J})$	6.6×10^{0}	2.4×10^{3}	3.7×10^{9}

TABLE 5.6
Numerical errors generated by alg_2 *for Example* $q = 10$ *in the three different cases.*

n	Case 1	Case 2	Case 3
100	1.1×10^{-15}	7.6×10^{-13}	3.2×10^{-6}
500	2.3×10^{-15}	8.4×10^{-13}	4.5×10^{-6}
1000	3.3×10^{-15}	8.7×10^{-13}	4.9×10^{-6}
$\kappa(\mathcal{J})$	6.6×10^{0}	2.4×10^{3}	3.7×10^{9}

TABLE 5.7
Numerical errors generated by alg_1 *for Example* $q = 20$ *in the three different cases.*

n	Case 1	Case 2	Case 3
100	1.3×10^{-15}	5.7×10^{-13}	8.0×10^{-4}
500	4.8×10^{-15}	5.6×10^{-13}	1.3×10^{-3}
1000	5.3×10^{-15}	5.6×10^{-13}	1.4×10^{-3}
$\kappa(\mathcal{J})$	7.5×10^{0}	1.6×10^{4}	1.6×10^{11}

TABLE 5.8
Numerical errors generated by alg_2 *for Example* $q = 20$ *in the three different cases.*

n	Case 1	Case 2	Case 3
100	1.6×10^{-15}	1.1×10^{-13}	2.0×10^{-4}
500	3.0×10^{-15}	1.3×10^{-13}	4.9×10^{-4}
1000	7.5×10^{-15}	1.7×10^{-13}	6.3×10^{-4}
$\kappa(\mathcal{J})$	7.5×10^{0}	1.6×10^{4}	1.6×10^{11}

are enlightened by a suitable additive decomposition of the rational matrix symbol. The computation of such a decomposition reduces to the solution of an associated Jury system. The two proposed algorithms differ in the choice of the generator set for the rank structures of the Toeplitz matrices. Numerical experiments show that the proposed approaches are numerically reliable and, whenever the Jury system is well-conditioned, the error in the computed eigenvalues is of the order of the norm of the input matrix.

REFERENCES

[1] P. ARBENZ AND G. H. GOLUB, *On the spectral decomposition of Hermitian matrices modified by low rank perturbations with applications*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 40–58.

[2] S. BARNETT *Note on the Bezoutian matrix*, SIAM J. Appl. Math., 22 (1972), pp. 84–86.

[3] D. A. BINI AND F. DI BENEDETTO, *Solving the generalized eigenvalue problem for rational Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 537–552.

[4] D. A. BINI AND V. Y. PAN, *Efficient algorithms for the evaluation of the eigenvalues of (block) banded Toeplitz matrices*, Math. Comp., 50 (1988), pp. 431–448.

[5] ———, *On the evaluation of the eigenvalues of a banded Toeplitz block matrix*, J. Complexity, 7 (1991), pp. 408–424.

[6] A. BÖTTCHER AND B. SILBERMANN, *Introduction to Large Truncated Toeplitz Matrices*, Springer, New York, 1999.

[7] J. R. BUNCH, C. P. NIELSEN, AND D. C. SORENSEN, *Rank-one modification of the symmetric eigenvalue problem*, Numer. Math., 31 (1978), pp. 31–48.

[8] S. CHANDRASEKARAN AND M. GU, *Fast and stable eigendecomposition of symmetric banded plus semiseparable matrices*, Linear Algebra Appl., 313 (2000), pp. 107–114.
[9] S. DELVAUX AND M. VAN BAREL, *The explicit QR-algorithm for rank structured matrices*, Technical Report TW459, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 2006.
[10] ———, *A QR-based solver for rank structured matrices*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 464–490.
[11] ———, *A Givens-weight representation for rank structured matrices*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1147–1170.
[12] ———, *A Hessenberg reduction algorithm for rank structured matrices*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 895–926.
[13] C. J. DEMEURE AND C. T. MULLIS, *A Newton-Raphson method for moving-average spectral factorization using the Euclid algorithm*, IEEE Trans. Acoust. Speech Signal Process., 38 (1990), pp. 1697–1709.
[14] P. DEWILDE AND A.-J. VAN DER VEEN, *Time-varying Systems and Computations*, Kluwer Academic Publishers, Boston, 1998.
[15] F. DI BENEDETTO, *Computing eigenvalues and singular values of Toeplitz matrices*, Calcolo, 33 (1996), pp. 237–248.
[16] ———, *Generalized updating problems and computation of the eigenvalues of rational Toeplitz matrices*, Linear Algebra Appl., 267 (1997), pp. 187–219.
[17] B. W. DICKINSON, *Solution of linear equations with rational Toeplitz matrices*, Math. Comp., 34 (1980), pp. 227–233.
[18] Y. EIDELMAN, L. GEMIGNANI, AND I. C. GOHBERG, *On the fast reduction of a quasiseparable matrix to Hessenberg and tridiagonal forms*, Linear Algebra Appl., 420 (2007), pp. 86–101.
[19] Y. EIDELMAN AND I. C. GOHBERG, *Inversion formulas and linear complexity algorithm for diagonal plus semiseparable matrices*, Comput. Math. Appl., 33 (1997), pp. 69–79.
[20] ———, *On a new class of structured matrices*, Integral Equations Operator Theory, 34 (1999), pp. 293–324.
[21] ———, *A modification of the Dewilde-van der Veen method for inversion of finite structured matrices*, Linear Algebra Appl., 343-344 (2002), pp. 419–450.
[22] ———, *Fast inversion algorithms for a class of structured operator matrices*, Linear Algebra Appl., 371 (2003), pp. 153–190.
[23] Y. EIDELMAN, I. C. GOHBERG, AND V. OLSHEVSKY, *The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order*, Linear Algebra Appl., 404 (2005), pp. 305–324.
[24] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.
[25] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Third ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[26] M. GU AND S. C. EISENSTAT, *A stable and efficient algorithm, for the rank-one modification of the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1266–1276.
[27] S. L. HANDY AND J. L. BARLOW, *Numerical solution of the eigenproblem for banded symmetric Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 205–214.
[28] P. HENRICI, *Applied and Computational Complex Analysis*, Vol. 1, John Wiley & Sons, New York, 1974.
[29] M. KAC, W. L. MURDOCK, AND G. SZEGÖ, *On the eigenvalues of certain Hermitian forms*, Arch. Ration. Mech. Anal., 2 (1953), pp. 767–800.
[30] P. G. MARTINSSON, V. ROKHLIN, AND M. TYGERT, *A fast algorithm for the inversion of general Toeplitz matrices*, Comput. Math. Appl., 50 (2005), pp. 741–752.
[31] V. ROKHLIN AND M. TYGERT, *Fast algorithms for spherical harmonic expansions*, SIAM J. Sci Comput., 27 (2006), pp. 1903–1928.
[32] H. R. SCHWARZ, *Handbook series linear algebra: Tridiagonalization of a symmetric band matrix*, Numer. Math., 12 (1968), pp. 231–241.
[33] W. F. TRENCH, *On the eigenvalue problem for Toeplitz band matrices*, Linear Algebra Appl., 64 (1985), pp. 199–214.
[34] ———, *On the eigenvalue problem for Toeplitz matrices generated by rational functions*, Linear Multilinear Algebra, 17 (1985), pp. 337–353.
[35] ———, *Numerical solution of the eigenvalue problem for symmetric rationally generated Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 291–303.
[36] ———, *Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 135–146.
[37] ———, *Spectral evolution of the eigenvalue problem for Hermitian Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 601–611.
[38] E. E. TYRTYSHNI KOV, *Mosaic ranks for weakly semiseparable matrices*, in Large-Scale Scientific Computations of Engineering and Environmental Problems II, M. Griebel, S. Margenov, and P. Y. Yalamov, eds., Notes Numer. Fluid Mech. 73, Vieweg, 2000, pp. 36–41.
[39] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *Matrix Computations and Semiseparable Matrices, Volume I: Linear Systems*, The Johns Hopkins University Press, Baltimore, MD, 2008.
[40] ———, *Matrix Computations and Semiseparable Matrices, Volume II: Spectral Methods*, The Johns Hopkins

University Press, Baltimore, MD, 2008.

[41] D. S. WATKINS, *Fundamentals of Matrix Computations*, Pure and Applied Mathematics, John Wiley & Sons, Inc., New York, 2002.

[42] N. L. ZAMARASHKIN, I. OSELEDETS, AND E. E. TYRTYSHNIKOV, *On the approximation of Toeplitz matrices by a sum of the circulant and low-rank matrix*, Dokl. Akad. Nauk, 406 (2006), pp. 602–603.