2008

Computations with quasiseparable polynomials and matrices

T. Bella
University of Rhode Island

Y. Eidelman

See next page for additional authors

Follow this and additional works at: https://digitalcommons.uri.edu/math_facpubs

Terms of Use
All rights reserved under copyright.

Citation/Publisher Attribution
Available at: https://doi.org/10.1016/j.tcs.2008.09.008

This Article is brought to you for free and open access by the Mathematics at DigitalCommons@URI. It has been accepted for inclusion in Mathematics Faculty Publications by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons@etal.uri.edu.
Computations with quasiseparable polynomials and matrices

T. Bella\textsuperscript{a,}\textsuperscript{*}, Y. Eidelman\textsuperscript{c}, I. Gohberg\textsuperscript{c}, V. Olshevsky\textsuperscript{b}

\textsuperscript{a} Department of Mathematics, University of Rhode Island, Kingston, RI 02881-0816, USA
\textsuperscript{b} Department of Mathematics, University of Connecticut, Storrs CT 06269-3009, USA
\textsuperscript{c} School of Mathematical Sciences, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Ramat-Aviv 69978, Israel

A R T I C L E I N F O

Keywords:
Semiseparable matrices
Quasiseparable matrices
Orthogonal polynomials
Szegö polynomials
Recurrence relations
QR iteration
Divide and conquer eigenvalue algorithms
Signal flow graphs
Digital filters
Björck–Pereyra algorithm
Traub algorithm

A B S T R A C T

In this paper, we survey several recent results that highlight an interplay between a relatively new class of quasiseparable matrices and univariate polynomials. Quasiseparable matrices generalize two classical matrix classes, Jacobi (tridiagonal) matrices and unitary Hessenberg matrices that are known to correspond to real orthogonal polynomials and Szegö polynomials, respectively. The latter two polynomial families arise in a wide variety of applications, and their short recurrence relations are the basis for a number of efficient algorithms. For historical reasons, algorithm development is more advanced for real orthogonal polynomials. Recent variations of these algorithms tend to be valid only for the Szegö polynomials; they are analogues and not generalizations of the original algorithms.

Herein, we survey several recent results for the “superclass” of quasiseparable matrices, which includes both Jacobi and unitary Hessenberg matrices as special cases. The interplay between quasiseparable matrices and their associated polynomial sequences (which contain both real orthogonal and Szegö polynomials) allows one to obtain true generalizations of several algorithms. Specifically, we discuss the Björck–Pereyra algorithm, the Traub algorithm, certain new digital filter structures, as well as QR and divide and conquer eigenvalue algorithms.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

An interplay between polynomials and structured matrices is a well-studied topic, see, e.g., [48,44–46] and many references therein. In the context of polynomial computations, typically matrices with Toeplitz, Hankel, Vandemone, and related structures were of interest.

Recently, a rather different class of quasiseparable matrices has been receiving a lot of attention.\textsuperscript{1} The problems giving rise to quasiseparable matrices as well as the methods for attacking them are somewhat different from those for Toeplitz and Hankel matrices. We start by indicating (in Sections 1.1–1.3) one of the differences between these familiar classes of structured matrices and the new one.

\textsuperscript{*} Corresponding author.
E-mail address: tombella@math.uri.edu (T. Bella).

\textsuperscript{1} Quasiseparable and semiseparable matrices are currently among the chief topics of research of several groups in Belgium (Van Barel et al.), Israel (Eidelman, Gohberg), Italy (Bini, Gemignani, Fasino, Mastronardi), the USA (Pan, Gu, Chandrasekaran, Olshevsky), etc. It is virtually impossible to survey in one paper all aspects of their research, and we refer to [19,20,9,23,17,56] among others as well as many references therein. In this survey paper we limit our scope to the interplay between this quasiseparable class of matrices and associated sequences of univariate polynomials.

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
1.1. Classical polynomial families and their moment matrices

Real orthogonal polynomials (including, for instance, Chebyshev, Legendre, Hermite, Laguerre polynomials) are polynomials that are orthogonal with respect to an inner product $\langle \cdot, \cdot \rangle$ defined on a real interval $[a, b]$, of the form

$$\langle p(x), q(x) \rangle = \int_a^b p(x)q(x)w^2(x)dx,$$

where $w^2(x)$ is some weight function, and such polynomials are classical [50,30]. They arise in a variety of problems in scientific computing such as numerical integration/Gaussian quadrature [51], discrete sine/cosine transforms [34], systems and control [18], and solving differential equations [33].

Numerous applications in signal processing [47], system theory and control [36], inverse scattering [37] give rise to other orthogonal polynomial sequences called the Szegö polynomials. Szegö polynomial sequences are orthogonal with respect to the inner product $\langle \cdot, \cdot \rangle$ defined via integration on the unit circle, i.e.,

$$\langle p(x), q(x) \rangle = \int_{-\pi}^{\pi} p(e^{i\theta})q(e^{i\theta})w^2(\theta)d\theta.\tag{1.2}$$

It is well known that the matrix elements $H$ corresponding to real orthogonal polynomials have a Hankel structure (constant values along anti-diagonals), and the moment matrices $T$ of Szegö polynomials have a Toeplitz structure (constant values along diagonals), displayed in

$$H = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_n \\ h_1 & h_2 & h_3 & \cdots & h_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_n & h_{n+1} & \cdots & \cdots & h_{2n-1} \end{bmatrix}, \quad T = \begin{bmatrix} t_0 & t_{-1} & t_{-2} & \cdots & t_{-n} \\ t_1 & t_0 & t_{-1} & \cdots & \vdots \\ t_2 & t_1 & t_0 & \cdots & t_{-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_n & \cdots & t_2 & t_1 & t_0 \end{bmatrix}.\tag{1.3}$$

Both of these moment matrices $H$ and $T$ are shift–invariant (i.e., they have constant values along their anti-diagonals and diagonals, respectively). This can be immediately deduced from the definition of their corresponding inner product $\langle \cdot, \cdot \rangle$ above, along with the definition $m_{ij} = \langle x^i, x^j \rangle$ for the moments (i.e., for the entries of the corresponding moment matrix $M = [m_{ij}]$). Indeed, in the real line case it follows from (1.1) that $m_{ij} = \int_a^b x^i x^j w^2(x)dx$ depends on the sum of the row and column indices. Hence the matrix $M$ has a Hankel structure.\footnote{The structure of the Toeplitz matrix is deduced similarly from (1.2) and the fact that on the unit circle we have $e^{i\theta} = e^{-i\theta} = e^{i\theta}$, so that each moment $m_{ij}$ now depends only on the difference of indices, which yields the Toeplitz structure.}

The shift–invariant structure of $H$ and $T$ implies that these two square arrays are structured, i.e., they are defined by only $O(n)$ parameters each.

1.2. Classical polynomial families and their recurrence matrices

Besides Hankel and Toeplitz, there are two more classes of matrices associated with real orthogonal and Szegö polynomials, namely tridiagonal and unitary Hessenberg matrices, respectively, displayed next:

$$T_n = \begin{bmatrix} \delta_1 & \gamma_2 & 0 & \cdots & 0 \\ \gamma_2 & \delta_2 & \gamma_3 & \ddots & \vdots \\ 0 & \gamma_3 & \delta_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_n & \delta_n \end{bmatrix}, \quad U_n = \begin{bmatrix} -\rho_0^* \rho_1 & -\rho_0^* \mu_1 \rho_2 & -\rho_0^* \mu_1 \mu_2 \rho_3 & \cdots & -\rho_0^* \mu_1 \mu_2 \cdots \mu_{n-1} \rho_n \\ -\rho_0 \rho_1 & -\rho_0 \mu_1 \rho_2 & -\rho_0 \mu_1 \mu_2 \rho_3 & \cdots & -\rho_0 \mu_1 \mu_2 \cdots \mu_{n-1} \rho_n \\ \mu_1 & -\rho_1 \rho_2 & -\rho_1 \mu_2 \rho_3 & \cdots & -\rho_1 \mu_2 \cdots \mu_{n-1} \rho_n \\ 0 & \mu_2 & -\rho_2 \rho_3 & \cdots & -\rho_2 \mu_3 \cdots \mu_{n-1} \rho_n \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_{n-1} & -\rho_{n-1} \rho_n \end{bmatrix},\tag{1.4}$$

where $\mu_{k,j} = \mu_k \cdot \mu_{k+1} \cdots \cdot \mu_j$. Similar to the matrices in (1.3), the two square arrays in (1.4) are also structured, i.e., they are defined by only $O(n)$ parameters each. For example, the parameters $\{\delta_k, \gamma_k\}_{k=1}^n$ defining $T_n$ are taken from the three-term recurrence relations,

$$r_k(x) = (x - \delta_k) r_{k-1}(x) - \gamma_k^2 r_{k-2}(x),\tag{1.5}$$

that real orthogonal polynomials $\{r_k(x)\}$ are known to satisfy [51]. Similarly, the parameters $\{\rho_k, \mu_k\}_{k=0}^n$ defining $U_n$ are taken from the so-called two-term recurrence relations (to be provided later in (2.4)) satisfied by Szegö polynomials. This justifies the nomenclature recurrence matrices used for $T_n$ and $U_n$.\footnote{The structure of the Toeplitz matrix is deduced similarly from (1.2) and the fact that on the unit circle we have $e^{i\theta} = e^{-i\theta} = e^{i\theta}$, so that each moment $m_{ij}$ now depends only on the difference of indices, which yields the Toeplitz structure.}
1.3. Generalizations. Displacement structure matrices, quasiseparable matrices, and related polynomials

Many nice results originally derived only for Hankel and Toeplitz moment matrices (e.g., fast Levinson and Schur algorithms, fast multiplication algorithms, explicit inversion formulas, etc.) have been generalized to the more general classes of Toeplitz-like and Hankel-like matrices and to the even more general class of displacement structure matrices (that includes not only Toeplitz and Hankel, but also Vandermonde and Cauchy matrices as special cases). We refer to [38,32,14,35,46,43,42] and many references therein for the displacement structure theory and the list of applications. Here we only mention the following fact that will be relevant in a moment. While, say, Toeplitz structure is immediately lost under either inversion or matrix multiplication, its displacement structure is preserved [38], a very useful property in the design of fast algorithms. The point is that displacement structure allows one to compress information about the $n^2$ matrix entries into only $\Theta(n)$ entries of the so-called generators, which leads to savings in storage and eventually to computational savings of at least an order of magnitude.

One can observe that, similar to moment matrices, the classes of tridiagonal and unitary Hessenberg matrices are also not closed under either inversion or matrix multiplication. So, it looks natural to ask about a “superclass” of matrices (i.e., about a counterpart of displacement structure matrices) that would be closed under both inversion and matrix multiplication, and would include both tridiagonal and unitary Hessenberg matrices as special cases.

Surprisingly, such a “moment matrices” pattern of generalizations was never mimicked in the study of recurrence matrices. In fact, the “superclass” in question, namely, the class of quasiseparable structure, indeed exists, but its study was originally motivated by applications in the theory of time-varying systems [19,20]. To sum up, a fresh classification shown in Table 1 seems to suggest a new direction for attacking problems for quasiseparable matrices.

Specifically, since quasiseparable matrices are of the “recurrence type”, it is of interest to study their so-called quasiseparable polynomials. The point is that historically, algorithm development is more advanced for real orthogonal polynomials. Recently, several important algorithms originally derived for real orthogonal polynomials have been carried over to the class of Szegö polynomials (however, such new algorithms tend to be valid only for the Szegö polynomials; they are analogues and not generalizations of the original algorithms).

Consider quasiseparable polynomials leads, in many cases, to true generalizations of many known results. In this survey paper we describe such generalizations for several areas displayed in the following figure.

---

3 Polynomials corresponding to semiseparable matrices (which is, along with tridiagonal and unitary Hessenberg matrices, another subclass of quasiseparable matrices) were recently studied in [24].
1.4. Example: Divide and conquer eigenvalue problem

As an example, we consider the problem of finding the roots of the $n$-th polynomial of a sequence of real orthogonal polynomials $\{r_k(x)\}$ satisfying the recurrence relations (1.5). As was mentioned in Section 1.2, the polynomials $\{r_k(x)\}$ correspond to tridiagonal matrices of the form $T_n$ shown in (1.4). The eigenvalues of $T_n$ are the desired roots of $r_n(x)$ (see Section 2.1 for details).

**Known algorithm: Real orthogonal polynomials.** One standard method for finding the eigenvalues of a tridiagonal method is divide and conquer method, due to Cuppen [15,12,13]. The basic idea of this method is that (i) a problem of size $n$ can be subdivided into problems with the same structure, but of smaller size (the divide step), and (ii) if we know the solution of the smaller problems, we can construct the solution to the problem of size $n$ (the conquer step). Then the original problem can be reduced recursively, until several sufficiently small problems can be very rapidly solved, and as solution of the original problem of size $n$ can be built upwards from them.

In the tridiagonal case, such an algorithm is based on the observation that a rank-one modification of four of the entries of the tridiagonal structure yields two smaller tridiagonal matrices, as in

$$ T = \begin{bmatrix} * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ 0 & * & * & * & 0 \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix} = \begin{bmatrix} * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ 0 & * & * & * & 0 \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix} + A = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} + A \tag{1.6} $$

where $\text{rank}(A) = 1$. This observation leads to the divide step. In the conquer step, one obtains the eigendecomposition of $T$ given $A$ and the eigendecompositions of $T_1$ and $T_2$. Together these steps allow calculation of the eigendecomposition of a tridiagonal matrix, and thus the roots of real orthogonal polynomials.

**“Carried over” algorithm: Szegö polynomials.** It was noticed by Gragg and Reichel in [29] that a similar algorithm was possible for computing the roots of Szegö polynomials. Szegö polynomials are known to be related to (almost) unitary Hessenberg matrices $U_n$ shown in (1.4) (see Section 2.1 for further details). As above, the eigenvalues of this matrix $U_n$ are the roots of the $n$th Szegö polynomial, and thus we seek an eigendecomposition of $U_n$. The divide and conquer algorithm of [29] uses the fact that such matrices admit the well-known Schur factorization, which is an expression of $H$ as a product of $n - 1$ Givens rotations $G_i(\rho_i)$.

$$ U_n = G_1(\rho_1)G_2(\rho_2) \cdots G_{n-1}(\rho_{n-1})G_n(\rho_n), \quad \text{where } G_j(\rho_j) = \text{diag}\{I_{j-1}, \begin{bmatrix} \rho_j & \mu_j \\ -\mu_j & \rho_j \end{bmatrix}, I_{n-j-1}\}, \tag{1.7} $$

see, e.g., [25]. The algorithm due to Gragg and Reichel reduces the eigendecomposition of $U_n$ of (1.7), to the ones for the matrices $U_1$ and $U_2$, where (assuming $n$ is even for convenience)

$$ U_1 = G_1(\rho_1)G_2(\rho_2) \cdots G_{n/2-1}(\rho_{n/2-1})G_{n/2}(\rho_{n/2}) $$

$$ U_2 = G_{n/2+1}(\rho_{n/2+1})G_{n/2+2}(\rho_{n/2+2}) \cdots G_{n-1}(\rho_{n-1})G_n(\rho_n). $$

In other words, by modifying one of the parameters to $\tilde{\rho}_{n/2}$ (corresponding to the rank-one modification $A$ in the tridiagonal case), they divide the unitary Hessenberg matrix into two smaller ones, which gives a divide step. This leads to a divide and conquer algorithm for such matrices, and allows one to compute the desired roots of Szegö polynomials.

In general, however, tridiagonal matrices do not admit factorizations (1.7), so this algorithm is not valid for tridiagonal matrices, as it is not a generalization of the previous algorithm.

**Generalized algorithm: Quasiseparable polynomials.** We consider instead the concept of an order-one quasiseparable matrix, defined as a matrix such that

$$ \max \text{rank} A_{12} = \max \text{rank} A_{21} = 1 $$

for all symmetric partitions of the form

$$ A = \begin{bmatrix} * & A_{12} \\ A_{21} & * \end{bmatrix}. $$

It is shown in Section 7 that a rank-one modification (generalizing the one in (1.6)) can be made to such a quasiseparable matrix that results in two smaller quasiseparable matrices as in the previous two divide and conquer algorithms above.

We next show why the new scheme generalizes Cuppen’s and Gragg-Reichel divide and conquer algorithms. By considering typical partitions $A_{21}$ of tridiagonal matrices and unitary Hessenberg matrices, we note that as both are upper Hessenberg, all such partitions have the form

$$ A_{21} = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix}. $$
for both tridiagonal and unitary Hessenberg matrices, which are all rank one. Similarly, considering typical partitions $A_{12}$ for tridiagonal and unitary Hessenberg matrices, we have

\[
A_{12} = \begin{bmatrix}
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\ast & 0 & \cdots & 0
\end{bmatrix}
\]

and

\[
A_{12} = \begin{bmatrix}
-\rho_1 \mu_{k-1} \cdots \mu_3 \mu_2 \mu_1 \rho_0^* \\
-\rho_k \mu_{k-1} \cdots \mu_3 \mu_2 \rho_1^* \\
-\rho_0 \mu_{k-1} \cdots \mu_3 \rho_2^*
\end{bmatrix},
\]

(1.8)

respectively. We can observe that both are rank one (each row of the latter are scalar multiples of the others). Hence both tridiagonal and unitary Hessenberg matrices are (1, 1)-quasiseparable (or $(H, 1)$-quasiseparable, to emphasize that the lower part is not only order-one quasiseparable, but zeros below the subdiagonal; i.e. upper Hessenberg).

Since the class of quasiseparable matrices contains as subclasses the classes of tridiagonal and unitary Hessenberg matrices, an algorithm formulated in terms of quasiseparable structure results in a generalization of the previous work, as opposed to carrying algorithms over for the new case only.

1.5. Main results

The previous example demonstrates our main point of the paper: the interplay between polynomials and structured matrices allows one to generalize algorithms (instead of carrying them over) by considering the superclass of quasiseparable matrices/polynomials. This general theme is repeated several times, such as

- Szegö polynomials lead to the so-called lattice digital filter structures. We describe more general semiseparable and quasiseparable filter structures (Section 3).
- Björck–Perérya algorithms were carried over from three-term-Vandermonde matrices to Szegö–Vandermonde matrices [2]. We generalize these algorithms to the quasiseparable case. (Section 4).
- Traub algorithms were carried over from three-term-Vandermonde matrices to Szegö–Vandermonde matrices [40,41]. We generalize these algorithms to the quasiseparable case (Section 5).
- Many eigenvalue algorithms involve exploiting the tridiagonal/Hessenberg structure of the problem (often after creating that very structure by means of Householder reflections). As illustrated in the example of this section, generalizations to the quasiseparable case are possible here as well (Sections 6 and 7).

2. Interplay between polynomials and classes of structured matrices

2.1. Relationship between polynomials and matrices. Classical cases

In this section, we establish a bijection between Hessenberg matrices and certain sequences of polynomials. Let $\mathcal{H}$ be the set of all upper strongly Hessenberg matrices ("strongly" means that $a_{i+1,i} \neq 0$ for $i = 1, \ldots, n-1$ and $a_{i,j} = 0$ for $i > j+1$), and $\mathcal{P}$ be the set of all polynomial sequences $\{r_k(x)\}$ satisfying $\deg r_k = k$ (with $r_0(x)$ an arbitrary constant function). For any upper strongly Hessenberg $n \times n$ matrix $A \in \mathcal{H}$, define the function $f$ via the relation

\[
f(A) = P, \quad P = \{r_k\}_{k=0}^n, \quad r_k(x) = \frac{1}{a_{2,1}a_{3,2} \cdots a_{k,k-1}} \det(xI - A)_{k \times k};
\]

(2.1)

that is, associate with each upper strongly Hessenberg matrix $H$ the polynomial sequence consisting of the (scaled) characteristic polynomials of the principal submatrices of $H$. It is clear that this provides a mapping from matrices to polynomial sequences, i.e. $f : \mathcal{H} \to \mathcal{P}$. It is further true that (except for the freedom in scaling the first and the last polynomials) this function is a bijection, as stated in the next proposition. For the proof and discussion, see [4,1].

**Proposition 2.1.** The function $f : \mathcal{H} \to \mathcal{P}$ defined in (2.1) is a bijection up to scaling polynomials $r_0$ and $r_n$.

In what follows we will need the following remark that makes the motivating examples of polynomials considered above more explicit. Both of its relations are well-known, see for instance [4,1] for details.

**Remark 2.2.** (i) The class of real orthogonal polynomials in (1.5) is related via (2.1) to the class of irreducible triadiagonal matrices in (1.4).
(ii) The class of Szegö polynomials (to be formally defined by (2.4)) is related via (2.1) to the class of (almost\(^4\)) unitary Hessenberg matrices; that is, those of the form

\[
H =\begin{bmatrix}
-\rho_0^* \rho_1 & -\rho_0^* \mu_1 \rho_2 & -\rho_0^* \mu_1 \mu_2 \rho_3 & \cdots & -\rho_0^* \mu_1 \mu_2 \cdots \mu_{n-1} \mu_n \\
\mu_1 & -\rho_1^* \rho_2 & -\rho_1^* \mu_2 \rho_3 & \cdots & -\rho_1^* \mu_2 \cdots \mu_{n-1} \mu_n \\
0 & \mu_2 & -\rho_2^* \rho_3 & \cdots & -\rho_2^* \cdots \mu_{n-1} \mu_n \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \mu_{n-1} & -\rho_{n-1}^* \rho_n
\end{bmatrix}
\]  

(2.2)

with \(\rho_0 = -1\), \(|\rho_k| < 1\), \(k = 1, \ldots, n - 1\), \(|\rho_n| \leq 1\), \(\mu_k = \begin{cases} \sqrt{1 - |\rho_k|^2} & \text{if } |\rho_k| < 1 \\ 1 & \text{if } |\rho_k| = 1 \end{cases}\).

**Remark 2.3.** As was mentioned above, a topic of recent interest has been to create analogues of results valid for real orthogonal polynomials that are valid for Szegö polynomials. Another class of polynomials for which results valid for real orthogonal polynomials have been extended (see [24]) is the class of semiseparable polynomials (to be formally introduced in Section 2.3), currently understood as those related via (2.1) to matrices of the form given in the next definition.

**Definition 2.4** ((H, 1)-Semiseparable Matrices). A matrix \(A\) is called (H, 1)-semiseparable if (i) it is upper strongly Hessenberg (i.e., upper Hessenberg with nonzero subdiagonal entries), and (ii) it is of the form \(A = B + \text{striu}(A_U)\) where \(A_U\) is rank-one and \(B\) is lower bidiagonal (\(\text{striu}(A_U)\) denotes the strictly upper triangular portion of the matrix \(A_U\), and corresponds to the MATLAB command \(\text{striu}(A_U, 1)\)).

2.2. (H, 1)-quasiseparable matrices and their subclasses

A main theme of this survey paper is to provide an alternative method of extending results valid for real orthogonal polynomials. Instead of deriving analogous algorithms, we consider a superclass of polynomials containing real orthogonal, Szegö, semiseparable, and other classes of polynomials, and derive algorithms valid for this class. To this end we will need to introduce first the following “superclass” of matrices.

**Definition 2.5** ((H, 1)-Quasiseparable Matrices). A matrix \(A = [a_{ij}]\) is called (H, 1)-quasiseparable (i.e., the lower part is sparse because \(A\) is upper Hessenberg, and the upper part is order 1 quasiseparable) if (i) it is upper strongly Hessenberg \((a_{i+1,j} \neq 0 \text{ for } i = 1, \ldots, n - 1 \text{ and } a_{ij} = 0 \text{ for } i > j + 1)\), and (ii) \(\max(\text{rank}A_{12}) = 1\) where the maximum is taken over all symmetric partitions of the form

\[
A = \begin{bmatrix}
* & A_{12} \\
* & *
\end{bmatrix}
\]

In Section 2.3 we will introduce the class of (H, 1)-quasiseparable polynomials as those related to (H, 1)-quasiseparable matrices via (2.1).

While both involve rank structure, the quasiseparable and semiseparable structures are quite different; in fact, semiseparability implies quasiseparability, but not conversely. Indeed, if from Definition 2.4,

\[
A = B + \text{striu}(A_U), \quad \text{rank}(A_U) = 1,
\]

then any symmetric partition of the form in Definition 2.5 will have an \(A_{12}\) element of rank one because \(A_U\) (from which \(A_{12}\) is entirely formed) has rank one. Thus an (H, 1)-semiseparable matrix is (H, 1)-quasiseparable.

Conversely, however, consider a tridiagonal matrix. Any symmetric partition \(A_{12}\) of a tridiagonal matrix as in Definition 2.5 will have at most one nonzero entry, and thus be rank one. Thus, tridiagonal matrices are (H, 1)-quasiseparable. However, they are not necessarily (H, 1)-semiseparable. In order to be (H, 1)-semiseparable, we must be able to complete the strictly upper triangular portion to a rank one matrix. That is, one must specify the “?” entries of

\[
\begin{bmatrix}
? & * & 0 & \cdots & 0 \\
? & ? & * & \cdots & \vdots \\
? & ? & ? & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
? & ? & \cdots & ? & *
\end{bmatrix}
\]

\(^4\text{A matrix } H \text{ is called } \alpha(n) \text{ almost unitary Hessenberg matrix provided } H = UD \text{ for a unitary Hessenberg matrix } U \text{ and diagonal matrix } D = \text{diag}[1, \ldots, 1, \rho_n]. \text{It is unitary except for possibly a scaling of the last column.}
where \( \star \) denotes a possibly nonzero entry, such that the entire matrix is rank one, which is not possible with nonzero values of the \( \star \) elements. Thus, tridiagonal matrices are not necessarily \((H, 1)\)-semiseparable, and so the inclusion of semiseparable inside quasiseparable is proper.

It is easy to verify (see, for instance, (1.8) or [4, 1] for more details) that the class of \((H, 1)\)-quasiseparable matrices also contains unitary Hessenberg matrices. In fact, more details of these and other inclusions are given in the following theorem and definitions.

**Theorem 2.6.** The classes of matrices defined in this section interact (i.e., include, intersect, are disjoint) as in the following figure:

The proof of this theorem can be found in [4, 1], and we next present the details of the subclasses introduced in the figure.

**Definition 2.7 ((H, 1)-Well-Free Matrices).**

- An \( n \times n \) matrix \( A = (A_{ij}) \) is said to have a **well** in column \( 1 < k < n \) if \( A_{ik} = 0 \) for \( 1 \leq i < k \) and there exists a pair \((i, j)\) with \( 1 \leq i < k \) and \( k < j \leq n \) such that \( A_{ij} \neq 0 \).

In words, a matrix has a well in column \( k \) if all entries above the main diagonal in the \( k \)-th column are zero, except if all entries in the upper-right block to the right of these zeros are also zeros, as shown in the following illustration:

- A \((H, 1)\)-quasiseparable matrix is said to be \((H, 1)\)-**well-free** if none of its columns \( k = 2, \ldots, n - 1 \) contain wells.

In Section 2.3 we will introduce the class of \((H, 1)\)-well-free polynomials as those related to \((H, 1)\)-well-free matrices via (2.1).

**Definition 2.8 (Truncated Unitary Hessenberg Matrices).** A unitary Hessenberg matrix of the form (2.2) is called \( m \)-truncated provided \( \rho_k \) satisfy \( \rho_2 \neq 0, \ldots, \rho_m \neq 0, \rho_{m+1} = \cdots = \rho_n = 0 \).

**Definition 2.9 (Bidiagonal-like Matrices).** A matrix \( A \) is called bidiagonal-like if (i) it is upper strongly Hessenberg, and (ii) it is of the form \( A = B + C \), with \( B \) a lower bidiagonal matrix, and \( C \) a matrix with at most one nonzero entry, and that entry is located in the first superdiagonal.

At the moment, new polynomial families have been introduced via association with particular matrix classes. In the next section this association is exploited to provide efficient recurrence relations for each of the new polynomial classes.
2.3. \((H, 1)\)-quasiseparable polynomials and their subclasses

As was mentioned in Remark 2.2, real orthogonal polynomials and Szegö polynomials are known to be related via (2.1) to tridiagonal and unitary Hessenberg matrices, respectively. These two facts imply that these polynomial classes satisfy sparse (e.g., three-term or two-term) recurrence relations. Moreover, the sparseness of these relations is the chief reason for the existence of a number of fast efficient algorithms.

In Sections 2.1 and 2.2, we defined several classes of matrices, e.g., \((H, 1)\)-quasiseparable, \((H, 1)\)-semiseparable, and \((H, 1)\)-well-free. One can expect that polynomials related to each of these classes via (2.1) should also satisfy some kind of sparse relations. Indeed, in this section we provide equivalent definitions for these classes of polynomials in terms of the recurrence relations satisfied. We begin with brief recalling the classical examples of real orthogonal and Szegö polynomials.

\textbf{Definition 2.10} (Real Orthogonal Polynomials and Three-term Recurrence Relations). A sequence of polynomials is real orthogonal if it satisfies the three-term recurrence relations

\[
\mathcal{r}_k(x) = (\alpha_k x - \delta_k) \mathcal{r}_{k-1}(x) - \gamma_k \cdot \mathcal{r}_{k-2}(x), \quad \alpha_k \neq 0, \gamma_k > 0. \tag{2.3}
\]

\textbf{Definition 2.11} (Szegö Polynomials and Two-term Recurrence Relations). A sequence of polynomials are Szegö polynomials \(\phi^k\) if they satisfy the two-term recurrence relations (with some auxiliary polynomials \(\phi^k\))

\[
\begin{bmatrix}
\phi_k(x) \\
\phi^k(x)
\end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix}
1 & -\rho_k^2 \\
-\rho_k & 1
\end{bmatrix} \begin{bmatrix}
\phi_{k-1}(x) \\
\phi^k_{k-1}(x)
\end{bmatrix},
\]

where the reflection coefficients \(\rho_k\) and complementary parameters \(\mu_k\) satisfy

\[|\rho_k| \leq 1, \quad \mu_k = \begin{cases} 
\sqrt{1 - |\rho_k|^2} & |\rho_k| < 1 \\
1 & |\rho_k| = 1.
\end{cases}\]

\textbf{Definition 2.12} ((H, 1)-Semiseparable Polynomials and Szegö-type Two-term Recurrence Relations). A sequence of polynomials \(\mathcal{r}_k(x)\) is called (H, 1)-semiseparable if it satisfies

\[
\begin{bmatrix}
\mathcal{G}_k(x) \\
\mathcal{r}_k(x)
\end{bmatrix} = \begin{bmatrix}
\alpha_k & \beta_k \\
\gamma_k & 1
\end{bmatrix} \begin{bmatrix}
\mathcal{G}_{k-1}(x) \\
(\delta_k x + \theta_k) \mathcal{r}_{k-1}(x)
\end{bmatrix},
\]

where \(\mathcal{G}_k\) are a sequence of auxiliary polynomials. The relations (2.5) are called Szegö-type two-term recurrence relations since they generalize the classical Szegö relations (2.4).

\textbf{Definition 2.13} ((H, 1)-Quasiseparable Polynomials and [EGO05]-type Two-term Recurrence Relations). A sequence of polynomials \(\mathcal{r}_k(x)\) is called (H, 1)-quasiseparable if it satisfies the following two-term recurrence relations,

\[
\begin{bmatrix}
\mathcal{G}_k(x) \\
\mathcal{r}_k(x)
\end{bmatrix} = \begin{bmatrix}
\alpha_k & \beta_k \\
\gamma_k & \delta_k x + \theta_k
\end{bmatrix} \begin{bmatrix}
\mathcal{G}_{k-1}(x) \\
\mathcal{r}_{k-1}(x)
\end{bmatrix},
\]

where \(\mathcal{G}_k\) are a sequence of auxiliary polynomials. The relations (2.6) are called [EGO05]-type two-term recurrence relations since they are a special case of another formula given in [22].

In order to provide a complete picture of the relationship between real orthogonal polynomials, Szegö polynomials, and the new proposed class of \((H, 1)\)-quasiseparable polynomials, we give the following theorem and definitions, in complete agreement with the corresponding classes of matrices in Theorem 2.6.

\textbf{Theorem 2.14} ([4,1]). The subsets of polynomials defined in this section interact (i.e., intersect, include, are disjoint) as in the following figure.
Remark 2.15. The detailed statement and proof of the above theorem can be found in [4,1], and it is based on the following fact that seems to be of interest by itself: the shape and configuration shown in the two figures of Theorems 2.6 and 2.14 are identical. This is a reflection of the already mentioned fact, proven in [4,1], that for each pair of identical classes we have the following: the polynomials described by any given rectangle in the figure of Theorem 2.14 are related via (2.1) to the matrices described by the corresponding rectangle in the figure of Theorem 2.6. More details on this will be given in Sections 2.4.2 and 2.4.3.

Similar to what was done after formulating Theorem 2.6 in Section 2.2, we provide complete definitions for all remaining classes of polynomials mentioned in Theorem 2.14.

Definition 2.16 ((H, 1)-Well-free Polynomials). A sequence of polynomials is called (H, 1)-well-free if it satisfies the general three-term recurrence relations

\[ r_k(x) = (\alpha_k x - \delta_k) \cdot r_{k-1}(x) - (\beta_k x + \gamma_k) \cdot r_{k-2}(x). \]  

Definition 2.17 (Truncated Szegő Polynomials). A sequence of Szegő polynomials is called \( m \)-truncated if the sequence satisfies the Geronimus-type three-term recurrence relations

\[
\phi_k^m(x) = \begin{cases} 
\frac{1}{\mu_0} & k = 0 \\
\frac{1}{\mu_1} x \phi_0^m(x) + \rho_1 \rho_0^* \phi_0^m(x) & k = 1 \\
\frac{1}{\mu_2} x \phi_1^m(x) - \frac{\mu_1 \mu_2}{\mu_2^2} \phi_0^m(x) & k = 2, \ \rho_1 = 0 \\
\frac{1}{\mu_2} x + \frac{\mu_2}{\mu_1} \phi_1^m(x) - \frac{\mu_1}{\mu_2} \frac{\mu_1}{\mu_2} x \phi_0^m(x) & k = 2, \ \rho_1 \neq 0 \\
\frac{1}{\mu_k} x + \frac{\mu_k}{\mu_{k-1}} \phi_{k-1}^m(x) - \frac{\mu_{k-1}}{\mu_k} \frac{\mu_k}{\mu_{k-1}} x \phi_{k-2}^m(x) & 2 < k \leq m \\
x \cdot \phi_{k-1}^m(x) & k > m.
\end{cases}
\]  

Definition 2.18 (Almost Factored Polynomials). A sequence of polynomials is called almost factored if it satisfies the bidiagonal-like three-term recurrence relations: for some \( j \in [1, n] \),

\[
r_k(x) = \begin{cases} 
(\alpha_k x - \delta_k) \cdot r_{k-1}(x) & k \neq j \\
(\alpha_{k-1} x - \delta_{k-1})(\alpha_k x - \delta_k) - \gamma_k \cdot r_{k-2}(x) & k = j.
\end{cases}
\]  

We also consider polynomials satisfying unrestricted three-term recurrence relations of the form

\[
r_k(x) = (\alpha_k x - \delta_k) r_{k-1}(x) - \gamma_k \cdot r_{k-2}(x), \quad \alpha_k \neq 0;
\]  

that is, the same recurrence relations as satisfied by real orthogonal polynomials, but without the restriction of \( \gamma_k > 0 \).
2.4. Sparse generators and an interplay between subclasses of \((H, 1)\)-quasiseparable matrices and \((H, 1)\)-quasiseparable polynomials

2.4.1. Key concept. Sparse generators for \((H, 1)\)-quasiseparable matrices

The sparse recurrence relations (2.6) allow us to “compress” the representation of \((H, 1)\)-quasiseparable polynomials; that is, instead of computations with the \(O(n^2)\) coefficients of the polynomials themselves, one can carry out computations with the \(O(n)\) recurrence relation coefficients.

The small ranks in off diagonal blocks of quasiseparable matrices allow one to compress the \(n^2\) entries of the matrices into a linear array in a similar way. It is possible to use only \(O(n)\) parameters, which we will denote generators, in place of the \(n^2\) entries of the matrix in algorithms.

**Theorem 2.19.** Let \(A\) be an \(n \times n\) matrix. Then \(A\) is an \((H, 1)\)-quasiseparable matrix if and only if there exists a set of \(6n - 6\) scalar parameters \(\{p_i, q_i, d_i, g_i, b_i, h_i\}\) for \(i = 1, \ldots, n - 1, j = 2, \ldots, n, k = 2, \ldots, n - 1,\) and \(l = 1, \ldots, n,\) such that

\[
A = \begin{pmatrix}
    d_1 & \cdots & g_1 b_1^x h_1 \\
    p_2 q_1 & \cdots & g_2 b_2^x h_2 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & p_n q_{n-1} b_n^x h_n
\end{pmatrix}
\]

where \(b_{ij}^x = b_{i+1} \cdots b_{j-1}\) for \(j > i + 1\)
and \(b_{ij}^x = 1\) for \(j = i + 1\)

(2.11)

**Remark 2.20.** At first glance, there is a certain redundancy in the above definition, i.e., \(2(n - 1)\) parameters \(\{p_i, q_i\}\) are used to define only \(n - 1\) entries of the first subdiagonal of \(A\). While it is correct (and one could just “drop” \(q_i\)'s), this notation is consistent with the standard notation in [22] for the more general (non-Hessenberg) case where both \(p_i\)’s and \(q_i\)’s are needed.

**Definition 2.21.** A set of elements \(\{p_i, q_i, d_i, g_i, b_i, h_i\}\) in the previous theorem are called **generators** of the matrix \(A\). Generators of a matrix are not unique, and this nonuniqueness extends nontrivially past the noted redundancy in \(p_i\)’s and \(q_i\)’s.

2.4.2. Complete characterization in the \((H, 1)\)-quasiseparable case

We next present two theorems from [4,1] that contain conversions formulas between the recurrence relation coefficients of (2.6) of Definition 2.13 and the generators of the corresponding \((H, 1)\)-quasiseparable matrices. These results will imply that \((H, 1)\)-quasiseparable matrices and \((H, 1)\)-quasiseparable polynomials provide a complete characterization of each other.

**Theorem 2.22** (Quasiseparable Generators \(\Rightarrow\) [EGO05]-Type Recurrence Relation Coefficients). Suppose an \(n \times n\) \((H, 1)\)-quasiseparable matrix \(A\) has generators \(\{p_k, q_k, d_k, g_k, b_k, h_k\}\). Then the polynomials \(R = \{r_k\}\) associated with \(A\) via (2.1) satisfies the recurrence relations

\[
\begin{bmatrix}
    F_k(x) \\
    r_k(x)
\end{bmatrix} = \frac{1}{p_{k+1}q_k} \begin{bmatrix}
    q_k p_k b_k & -q_k g_k \\
    p_k h_k & x - d_k
\end{bmatrix} \begin{bmatrix}
    F_{k-1}(x) \\
    r_{k-1}(x)
\end{bmatrix}
\]

(2.12)

for some sequence of auxiliary polynomials \(\{F_k\}\).

**Theorem 2.23** ([EGO05]-Type Recurrence Relation Coefficients \(\Rightarrow\) Quasiseparable Generators). Suppose that a sequence of \(n + 1\) polynomials \(R\) satisfies the recurrence relations

\[
\begin{bmatrix}
    G_k(x) \\
    r_k(x)
\end{bmatrix} = \begin{bmatrix}
    \alpha_k & \beta_k \\
    \gamma_k & \delta_k x + \theta_k
\end{bmatrix} \begin{bmatrix}
    G_{k-1}(x) \\
    r_{k-1}(x)
\end{bmatrix}
\]

Then the \((H, 1)\)-quasiseparable matrix with generators

\[
p_k = 1, \quad k = 2, \ldots, n, \quad q_k = 1/\delta_k, \quad k = 1, \ldots, n - 1, \quad d_k = -\theta_k/\delta_k, \quad k = 1, \ldots, n
\]

\[
g_k = \beta_k, \quad k = 1, \ldots, n - 1, \quad b_k = \alpha_k, \quad k = 2, \ldots, n - 1, \quad h_k = -\gamma_k/\delta_k, \quad k = 2, \ldots, n
\]

corresponds to \(R\) via (2.1).
Table 2
Characterizations of subclasses of \((H, 1)\)-quasiseparable matrices via related polynomials

<table>
<thead>
<tr>
<th>Class of matrices</th>
<th>Class of polynomials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Irreducible tridiagonal matrices</td>
<td>Real orthogonal polynomials (2.3)</td>
</tr>
<tr>
<td>Unitary Hessenberg matrices</td>
<td>three-term (2.4)</td>
</tr>
<tr>
<td>((H, 1))-semiseparable matrices</td>
<td>two-term (2.4)</td>
</tr>
<tr>
<td>Definition 2.4</td>
<td>((H, 1))-semiseparable polynomials (2.5)</td>
</tr>
<tr>
<td>Definition 2.7</td>
<td>((H, 1))-well-free polynomials (2.7)</td>
</tr>
<tr>
<td>((H, 1))-quasiseparable matrices</td>
<td>([\text{EGO05}])-type 2-term (2.6)</td>
</tr>
</tbody>
</table>

Table 3
Defining several important classes of matrices via their \((H, 1)\)-quasiseparable generators

<table>
<thead>
<tr>
<th>Matrix class</th>
<th>(p_i)</th>
<th>(q_i)</th>
<th>(d_i)</th>
<th>(g_i)</th>
<th>(b_i)</th>
<th>(h_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tridiagonal</td>
<td>1 (1/\alpha_i)</td>
<td>(\delta_i/\alpha_i)</td>
<td>(\gamma_{i+1}/\alpha_{i+1})</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Unitary Hessenberg</td>
<td>1 (\mu_i)</td>
<td>(-\rho_{i-1}^+ \rho_i)</td>
<td>(-\rho_{i-1}^+ \mu_i)</td>
<td>(\mu_k)</td>
<td>(\rho_i)</td>
<td></td>
</tr>
<tr>
<td>((H, 1))-semiseparable</td>
<td>(\neq 0) (\neq 0)</td>
<td>*</td>
<td>*</td>
<td>(\neq 0)</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>((H, 1))-well-free</td>
<td>(\neq 0) (\neq 0)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(\neq 0)</td>
<td></td>
</tr>
<tr>
<td>Bidiagonal-like</td>
<td>(\neq 0) (\neq 0)</td>
<td>*</td>
<td>(\neq 0) at most once</td>
<td>0</td>
<td>(\neq 0)</td>
<td></td>
</tr>
</tbody>
</table>

Table 4
Conversion formulas of the type “Recurrence relation coefficients \(\Leftrightarrow\) quasiseparable generators”

<table>
<thead>
<tr>
<th>Polynomials</th>
<th>(p_k)</th>
<th>(q_k)</th>
<th>(d_k)</th>
<th>(g_k)</th>
<th>(b_k)</th>
<th>(h_k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real orth. (2.3)</td>
<td>1</td>
<td>(1/\alpha_k)</td>
<td>(\delta_k/\alpha_k)</td>
<td>(\gamma_k/\alpha_k)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Szegö (2.4)</td>
<td>1 (\mu_k)</td>
<td>(-\rho_k \rho_{k-1}^+)</td>
<td>(\rho_{k-1}^+)</td>
<td>(\mu_{k-1})</td>
<td>(-\mu_{k-1} \rho_k)</td>
<td></td>
</tr>
<tr>
<td>Gen. 3-term (2.7)</td>
<td>1 (1/\alpha_k)</td>
<td>(\frac{\delta_k}{\alpha_k} + \frac{\rho_k}{\alpha_k - \gamma_k})</td>
<td>(\frac{\mu_k}{\alpha_k} + \frac{\gamma_k}{\alpha_k - \gamma_k})</td>
<td>(\frac{\mu_k}{\alpha_k} + \frac{\gamma_k}{\alpha_k - \gamma_k})</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Szegö-type (2.5)</td>
<td>1</td>
<td>(1/\delta_k)</td>
<td>(-\frac{\delta_k}{\delta_k - \gamma_k})</td>
<td>(\beta_{k-1})</td>
<td>(\alpha_{k-1} - \beta_{k-1} \gamma_{k-1})</td>
<td>(\frac{\delta_k}{\delta_k - \gamma_k})</td>
</tr>
<tr>
<td>([\text{EGO05}])-type (2.6)</td>
<td>1</td>
<td>(1/\delta_k)</td>
<td>(-\frac{\delta_k}{\delta_k - \gamma_k})</td>
<td>(\beta_k)</td>
<td>(\alpha_k)</td>
<td>(\delta_k/\delta_k - \gamma_k)</td>
</tr>
</tbody>
</table>

2.4.3. Complete characterization in the \((H, 1)\)-semiseparable and \((H, 1)\)-well-free cases

The two theorems of the previous section imply that \((H, 1)\)-quasiseparable polynomials (2.6) and \((H, 1)\)-quasiseparable matrices (2.11) completely characterize each other, being related via (2.1). The results included in the Table 2 indicate that similar complete characterizations exist for all other classes of matrices and polynomials of Theorems 2.6 and 2.14.

In the \((H, 1)\)-quasiseparable case the complete characterization result was established in the previous subsection via conversion formulas of the type “recurrence relation coefficients \(\Leftrightarrow\) \((H, 1)\)-quasiseparable generators”. Similar results for all other classes in Table 2 can be obtained via the following two-step procedure.

- Recall that Theorem 2.6 asserts that all matrix classes in question are subclasses of \((H, 1)\)-quasiseparable matrices. Hence one should first obtain for each of these matrix classes full descriptions in terms of restrictions on their \((H, 1)\)-quasiseparable generators. These restrictions are given in Table 3, where a • indicates that no restriction is placed on that particular generator.

In what follows, we present new feedforward digital filter structures based on \((H, 1)\)-quasiseparable polynomials, and in the following three sections we present generalizations to the class of \((H, 1)\)-quasiseparable matrices of the classical algorithms of Traub, Björck–Pereyra, and Cuppen. For each generalization, one can use the special restrictions of Table 3 to reduce the generalized version of the algorithm to the classical version. Thus \((H, 1)\)-quasiseparable matrices provide a unifying approach to solving all of the respective computational problems.

- Secondly, one needs conversion formulas of the type “recurrence relation coefficients \(\Leftrightarrow\) restricted \((H, 1)\)-quasiseparable generators”. We provide next the desired conversions only in one direction (since it will be used in what follows), and we refer to [4,1] for the conversion in another direction and all details.

In what follows we present algorithms based on \((H, 1)\)-quasiseparable polynomials, and the input for these algorithms will be the generators of the corresponding \((H, 1)\)-quasiseparable matrix. If one would need to apply these algorithms using not generators of the corresponding matrix but the recurrence relations coefficients as input, one way to do so is to precede the algorithms with conversions specified in Table 4.

All of the results of this section were recently extended to the more general \((H, m)\)-quasiseparable case in [8].
Table 5

<table>
<thead>
<tr>
<th>Building blocks of signal flow graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adder</td>
</tr>
<tr>
<td>Gain</td>
</tr>
<tr>
<td>Delay</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>r(x)</th>
<th>p(x) + r(x)</th>
<th>p(x)</th>
<th>αp(x)</th>
<th>xp(x)</th>
</tr>
</thead>
</table>

- Implements polynomial addition.
- Implements scalar multiplication.
- Implements multiplication by \(x\).

Fig. 1. Implementation of the polynomial (3.13) expressed by a signal flow graph.

Fig. 2. Markel-Grey filter structure: Signal flow graph to realize the Szegő polynomials using two-term recurrence relations (2.4).

3. New quasiseparable digital filter structures

3.1. Filter structures and the Markel–Grey filter structure

Complementing the algebraic descriptions of polynomials, particularly those determined by sparse recurrence relations, one can consider the corresponding signal flow graphs. Briefly, the goal is to build a device to implement, or realize, a polynomial, using devices that implement the algebraic operations of polynomial addition, multiplication by \(x\), and scalar multiplication. These building blocks are shown next in Table 5.

As an example, an easy way to implement a polynomial expressed in the monomial basis by the coefficients \(\{P_0, P_1, P_2, P_3\}\) via

\[
p(x) = P_0 + P_1x + P_2x^2 + P_3x^3
\]  

(3.13)

by a signal flow graph is shown in Fig. 1.

An important result in signal processing is the Markel–Grey filter design, which realizes polynomials using the Szegő polynomials as a basis. The filter design uses the two-term recurrence relations (2.4), which gives the ladder structure shown in Fig. 2.

In view of Theorem 2.14, it is natural to consider the generalized filter structures of the next section.

3.2. New filter structures

We obtained the following generalizations of the important Markel–Grey filter structure. Specifically, the recurrence relations (2.5) can be realized by the semiseparable filter structure depicted in Fig. 3, and the recurrence relations (2.6) lead to the quasiseparable filter structure, depicted in Fig. 4.
Remark 3.1. The quasiseparable filter structure is a single filter structure that can realize both real orthogonal polynomials and Szegő polynomials, as well as the more general case of \((H, 1)\)-quasiseparable sequences. Such a single implementation allows, among other things, a device to be built that can realize both real orthogonal polynomials and Szegő polynomials without the need for multiple devices.

The essential tools in deriving the results of the previous sections are \((H, 1)\)-quasiseparable matrices and their connection to \((H, 1)\)-quasiseparable polynomials and the corresponding recurrence relations. In the next section, we give more details about this relationship.

4. Fast Björck–Pereyra linear sequence solver

4.1. Classical Björck–Pereyra algorithm and its extensions

The problem of solving Vandermonde linear systems, systems \(V(x)a = f\) with

\[
V(x) = \begin{bmatrix}
1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\
: & : & : & \cdots & : \\
1 & x_n & x_n^2 & \cdots & x_n^{n-1}
\end{bmatrix},
\]

is classical, with applications in interpolation, coding theory, etc. It is known that Vandermonde matrices are extremely ill-conditioned [53], and hence solving linear systems involving \(V(x)\) using Gaussian elimination (GE) can result in (i) loss of forward accuracy. Additionally, GE needs (ii) large \(n^2\) storage, and (iii) it is expensive, using \(\Theta(n^3)\) flops. In 1970, Björck and Pereyra introduced a fast algorithm for solving Vandermonde linear systems which was better than GE in every sense. (i) It often resulted in perfectly accurate solutions [10], (ii) it needs only \(\Theta(n)\) storage, and (iii) it is fast, using only \(\Theta(n^2)\) flops.\(^5\) The Björck–Pereyra algorithm is based on the following factorization of the inverse of a Vandermonde matrix:

\[
V(x)^{-1} = U_1 \cdots U_{n-1} \cdot L_{n-1} \cdots L_1,
\]

with certain bidirectional matrices \(U_k, L_k\). The exact form of these bidiagonal matrices is not important at the moment; we only mention that their entries need not be computed; they are readily available in terms of the nodes \(\{x_i\}\) defining \(V(x)\). Björck and Pereyra used (4.2) to compute \(a = (V(x))^{-1}f\) very efficiently. Indeed, the bidiagonal structure of \(U_k, L_k\) obviously implies that only \(\Theta(n^2)\) operations are needed. Further, since the entries of \(U_k, L_k\) are readily defined by \(\{x_i\}\), only \(\Theta(n)\) storage is needed.

\(^5\) It is easy to solve a Vandermonde system in \(\Theta(n \log n)\) flops but such superfast algorithms are typically totally inaccurate already for 15 \(\times\) 15 matrices.
The speed and accuracy of the classical Björck–Pereyra algorithm attracted much attention, and as a result the algorithm has been generalized to several special cases of polynomial–Vandermonde matrices of the form

\[ V_k(x) = \begin{bmatrix} r_0(x_1) & r_1(x_1) & \cdots & r_{n-1}(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ r_0(x_n) & r_1(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix}, \quad \text{for } k \geq 0 \tag{4.3} \]

namely to those specified in Table 6.

The algorithms of [49,31,2] are not generalizations of the original Björck–Pereyra [10] algorithm but analogs of it.


A new Björck–Pereyra-type algorithm, obtained recently in [3,1], uses the superclass of quasiseparable polynomials and hence it is a generalization of all of previous work listed in Table 6. The new algorithm is based on the following theorem.

**Theorem 4.1.** Let \( V_k(x) \) be a polynomial–Vandermonde matrix of the form (4.3) for a sequence of \((H, 1)\)-quasiseparable polynomials \( R \) given by the generators \( \{p_j, q_j, d_j, g_j, f_j, h_j\} \) of the corresponding \((H, 1)\)-quasiseparable matrix \( A \) in (2.11). Then

\[ V_k(x)^{-1} = U_k \cdot \begin{bmatrix} I_{n-2} & L_{n-2} & \cdots & L_{n-2} \\ U_{n-1} & L_{n-1} & \cdots & L_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ U_2 & L_2 & \cdots & L_2 \end{bmatrix} \cdot L_1, \quad \text{for } k > 0 \tag{4.4} \]

\[ U_k = \begin{bmatrix} a_0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad L_k = \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ -1 \\ 1 \end{bmatrix}, \quad \text{where } A_{(n-k)\times(n-k)} \text{ is the leading } (n-k) \times (n-k) \text{ submatrix of } A \text{ in (2.11), and } a_0 = r_0(x) \text{ and for } k > 0 \text{ the number } a_k \text{ is the ratio of the leading coefficients of } r_k(x) \text{ and } r_{k-1}(x). \tag{4.5} \]

Remark 4.2. The overall cost for solving \( V_k(x) a = f \) via (4.4) is \( O(n^2) \) operations, since matrices \( L_k, U_k \) can be multiplied by a vector in \( O(n) \) operations each. For \( L_k \) it follows immediately from its sparsity. For \( U_k \) the linear cost of matrix-vector multiplication is possible thanks to the quasiseparable structure of its block \( A_{(n-k)\times(n-k)} \), see, e.g., [3,1] for details.

At first glance, the \( O(n^2) \) complexity may not seem optimal. As we mentioned, however, even for the well-studied simplest Vandermonde matrices in (4.1), all known algorithms with lower complexity are quite inaccurate in numerical computations over \( \mathbb{R} \) and \( \mathbb{C} \) (although in applications over finite fields they make perfect sense).

**Remark 4.3.** The input of the new algorithm based on (4.4) is a set of generators of the corresponding \((H, 1)\)-quasiseparable matrix \( A \) in (2.11). Hence it includes, as special cases, all the algorithms listed in Table 6. Indeed, in order to specify the new algorithm to a particular special case, one needs just to use its special generators listed in Table 3. Furthermore, in the case where the polynomials \( \{r_k(x)\} \) involved in (4.3) are given not by a generator, but by any of the recurrence relations of Section 2.3, one can simply precede the algorithm with the conversions of Table 4.

5. Fast Traub-like inversion algorithm

5.1. The classical Traub algorithm and its extensions

As mentioned in the previous section, numerical computations with classical Vandermonde matrices in (4.1) can be problematic due to their large condition numbers [53], and the problem of inversion of such matrices using Gaussian elimination suffers from the same problems as described in the previous section for solving linear systems. Gaussian elimination (i) cannot provide forward accuracy, and (ii) is expensive, requiring \( O(n^3) \) flops.
Table 7

<table>
<thead>
<tr>
<th>Polynomial system $R$</th>
<th>Fast $O(n^2)$ inversion algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monomials</td>
<td>Traub [52]</td>
</tr>
<tr>
<td>Chebyshev polynomials</td>
<td>Gohberg–Olshevsky [27]</td>
</tr>
<tr>
<td>Real orthogonal polynomials</td>
<td>Calvetti–Reichel [16]</td>
</tr>
<tr>
<td>Szegö polynomials</td>
<td>Olshevsky [40, 41]</td>
</tr>
</tbody>
</table>

Fast $O(n^2)$ inversion algorithms.

There is a fast Traub algorithm [52] that computes the entries of $V(x)^{-1}$ in only $O(n^2)$ operations, but it is known to be inaccurate as well. Fortunately, it was observed in [28] that a minor modification of the original Traub algorithm typically results in a very good forward accuracy.

The Traub algorithm has been generalized to polynomial–Vandermonde matrices of the form (4.3), namely to those specified in Table 7.

The algorithms of [27, 16, 40] are not generalizations of the original Traub [52] algorithm but its analogs.

5.2. A true generalization. New Traub-type algorithm for quasiseparable-Vandermonde matrices

A new Traub-type algorithm, obtained recently in [7, 1], uses the superclass of quasiseparable polynomials and hence it is a generalization of all of previous work listed in Table 7. The new algorithm is based on the following theorem.

**Theorem 5.1.** Let $V_R(x)$ be a polynomial–Vandermonde matrix of the form (4.3) for a sequence of $(H, 1)$-quasiseparable polynomials $R = \{r_k(x)\}$ given by the generators $\{p_j, q_l, d_i, g_i, b_i, h_l\}$ of the corresponding $(H, 1)$-quasiseparable matrix $A$ in (2.11). Then

$$V_R(x)^{-1} = \tilde{I} \cdot V^T_R(x) \cdot \text{diag}(c_1, \ldots, c_n),$$

where

$$\tilde{I} = \begin{bmatrix}
0 & \cdots & 0 & 1 \\
\vdots & \ddots & \vdots & \vdots \\
0 & \cdots & 1 & 0 \\
1 & 0 & \cdots & 0
\end{bmatrix}, \quad c_i = \prod_{k=1, k\neq i}^{n} (x_k - x_i)^{-1},$$

and $\hat{R} = \{\hat{r}_k(x)\}$ (defining the matrix $V_{\hat{R}}(x)$ in (5.1)) is the sequence of what are called the associated (or generalized Horner) polynomials that satisfy the following recurrence relations:

$$
\begin{bmatrix}
\hat{f}_k(x) \\
\hat{r}_k(x)
\end{bmatrix}
= \begin{bmatrix}
1 & \begin{bmatrix}
p_{n-k+1}q_{n-k+1}h_{n-k+1} \\
p_{n-k+1}g_{n-k+1}h_{n-k+1} \\
x - d_{n-k+1}
\end{bmatrix} \\
q_{n-k+1}h_{n-k+1} & 0
\end{bmatrix}
\begin{bmatrix}
\hat{f}_{k-1}(x) \\
\hat{r}_{k-1}(x)
\end{bmatrix}
+ \begin{bmatrix}
p_{n-k} \\
q_{n-k}h_{n-k+1}
\end{bmatrix}
\begin{bmatrix}
0 \\
\hat{r}_{n-k}(x)
\end{bmatrix}
$$

(5.3)

and $\hat{R}_i = \{\hat{r}_k(x)\}$ (defining the matrix $V_{\hat{R}_i}(x)$ in (5.1)) is the sequence of what are called the associated (or generalized Horner) polynomials that satisfy the following recurrence relations:

$$
\sum_{k=1}^{n} (x - x_k) = P_0 r_0(x) + P_1 r_1(x) + \cdots + P_{n-1} r_{n-1}(x) + P_n r_n(x),$$

(5.4)

**Remark 5.2.** The overall cost for computing $V_R(x)^{-1}$ via (5.1) is $O(n^2)$ operations. Indeed, $\tilde{I}$ is just a permutation, and the diagonal matrix $\text{diag}(c_1, \ldots, c_n)$ can be formed via (5.2) in $1.5n^2$ operations. Hence the cost of inversion is dominated by forming the matrix $V_{\hat{R}}$, which is a polynomial Vandermonde matrix of the form (4.3), but with associated (generalized Horner) polynomials $\{\hat{r}_k(x)\}$ in place of the original $\{r_k(x)\}$. Since [7, 1] contains an $O(n^2)$ algorithm for computing $P_k$'s, the recurrence relations (5.3) allow us to form the matrix $V_{\hat{R}}$ in $O(n^2)$ operations which results in the overall cost of $O(n^2)$ operations for the entire algorithm.

**Remark 5.3.** As in Remark 4.3, note that the input of the new algorithm based on (5.1) is a set of generators of the corresponding $(H, 1)$-quasiseparable matrix $A$ in (2.11). Hence it includes, as special cases, all the algorithms listed in Table 7. Indeed, in order to specify the new algorithm to a particular special case, one needs just to use its special generators listed in Table 3.

To apply the algorithm when the polynomials $\{r_k(x)\}$ involved in (4.3) are given not by a generator, but by any of the recurrence relations of Section 2.3, one can simply precede the new Traub-type algorithm by using the conversions of Table 4.
Remark 5.4. Comparing the recurrent relations (2.12) for the original \((H, 1)\)-quasiseparable polynomials \(\{r_k(x)\}\) (involved in \(V_R(x)\)) with the recurrent relations (5.3) for the associated polynomials \(\{\tilde{r}_k(x)\}\) (needed to form \(V_R(x)^{-1}\)) we see the following two differences. (i) The relations (5.3) have almost the same form as (2.12), but they flip the order of indexing generators. (ii) As opposed to (2.12) the relations (5.3) contain a certain “perturbation term”.

In fact, along with (5.3) one can find in [7,1] two more versions of the new Traub-like algorithm. The version based on (5.3) can be called \([EGO05]\)-type version since (5.3) is a perturbed counterpart of the \([EGO05]\)-type relations (2.12). The two other versions in [7,1] are based on perturbed counterparts of Szegö-type and of general three-term recurrence relations. As opposed to the algorithm based on (5.3), the two other versions can be used under certain (mild) limitations.

Finally, the Traub-like algorithm described in this section has been recently extended to the more general \((H, m)\)-quasiseparable case in [5,1].

6. Eigenvalue problems: QR algorithms

6.1. Motivation

It is customary to compute the eigendecomposition of a symmetric matrix in two steps: (i) reduce the matrix to a similar tridiagonal matrix by means of Householder transformations (unitary reflections); (ii) run the QR algorithm on this tridiagonal matrix. For a given (or reduced in step (i)) matrix \(A^{(1)}\) one computes the QR factorization \(A^{(1)} = QR\) (where \(Q\) is orthogonal and \(R\) is upper triangular). The next iterant \(A^{(2)} = RQ + \sigma_1 I = QR\) is obviously similar to \(A^{(1)}\). It is well-known (see, e.g., [51] and the references therein) that with the right choice of shifts \(\sigma_k\) the sequence \(\{A^{(k)}\}\) rapidly converges to a diagonal matrix thus providing the desired eigenvalues. The point is that the tridiagonal structure of \(A^{(1)}\) is preserved, under QR iterations which makes the above scheme fast and practical.

This process has been recently generalized to semiseparable matrices in [9,11,54,55]; but again, these methods are not generalizations of the classical tridiagonal version, rather its analogs. The QR iteration algorithm for arbitrary order symmetric quasiseparable matrices was derived in [21] (non-Hessenberg quasiseparable matrices will be defined below). As opposed to its predecessors, the algorithm of [21] is a true generalization of the classical tridiagonal QR iteration algorithms as well as of its semiseparable counterparts.

In this section we (i) describe how to precede the QR iteration algorithm of [21] with a method of reduction of a general symmetric matrix to a quasiseparable form (that admits, similarly to tridiagonal matrices, sparse generator representations). Two comments are due. First, there is a freedom in this reduction method. Hence, there is, in fact, a “family” of quasiseparable matrices that can be obtained as a result, all similar to the original matrix. One member of this family is simply the “good old” tridiagonal matrix, indicating that our reduction scheme is a true generalization of the classical Householder method. Secondly, as we will see in a moment, the structure of the reduced matrix can be better captured not by a general quasiseparable representation, but rather by a more flexible out-of-band quasiseparable representation. The latter option allows one to decrease the order of quasiseparability and hence to reduce the required storage. (ii) We observe that (similarly to tridiagonal structure) the out-of-band quasiseparable structure is inherited under QR iterations. This observation was exploited in [6,1] to derive a generalization of the algorithm of [21] that applies to the more general out-of-band quasiseparable matrices.

This seems to be currently the most general algorithm of this type. Once again, the overall algorithm has a freedom in choosing certain parameters, so that it is actually a family of algorithms that included the classical Householder QR algorithm as a special case.

6.2. Reduction to tridiagonal form via Householder reflections

Before describing this new approach, we give a brief overview of the classical reduction to tridiagonal form. Recall that for any two nonzero, distinct vectors \(x, y \in \mathbb{C}^n\), there exists a Householder reflection \(P\) such that \(Px = \alpha y\) for some constant \(\alpha\), and \(P = P^{-1}\). In particular, for any vector \(x\), one can always find a matrix \(P\) such that

\[
P x = \begin{bmatrix}
\alpha & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 
\end{bmatrix}
\]

for some \(\alpha\). This fact can be used to reduce a given symmetric matrix to a similar tridiagonal matrix, implementing (i) above. We illustrate the reduction on the following \(4 \times 4\) example:

\[
A = \begin{bmatrix}
a_{11} & a_{21} & a_{31} & a_{41} \\
a_{21} & a_{22} & a_{32} & a_{42} \\
a_{31} & a_{32} & a_{33} & a_{34} \\
a_{41} & a_{42} & a_{43} & a_{44} 
\end{bmatrix}.
\]
Let $P_1$ be such that

$$
\begin{bmatrix}
  a_{21} \\
  a_{31} \\
  a_{41}
\end{bmatrix}
= \begin{bmatrix}
  * \\
  0 \\
  0
\end{bmatrix},
$$

(6.1)

and then since $A$ is symmetric, we have

$$
\begin{bmatrix}
  1 & 0 \\
  0 & P_1
\end{bmatrix}
\begin{bmatrix}
  a_{11} & a_{21} & a_{31} & a_{41} \\
  a_{21} & * & a_{31} & * \\
  a_{31} & a_{32} & * & a_{41} \\
  a_{41} & a_{42} & a_{42} & *
\end{bmatrix}
\begin{bmatrix}
  1 & 0 \\
  0 & P_1
\end{bmatrix}
= \begin{bmatrix}
  a_{11} & * & 0 & 0 \\
  * & 0 & P_1A_1P_1 & *
\end{bmatrix},
$$

and repeating this procedure on $P_1A_1P_1$, one gets a symmetric tridiagonal matrix that is similar to the original symmetric matrix $A$.

We suggest a similar approach, but instead of a reduction to tridiagonal form, we give a reduction to a slightly modified version of quasiseparable matrices for which this structure is particularly suited.

### 6.3. Reduction to (out-of-band) quasiseparable form

In this section we start with considering a more general class of quasiseparable matrices than $(H, 1)$-quasiseparable matrices defined in Definition 2.5.

**Definition 6.1 (Quasiseparable Matrices).** A matrix $A$ is called $(n_L, n_U)$-quasiseparable if

$$
\text{max}\{\text{rank } A_{21}\} = n_L, \quad \text{and} \quad \text{max}\{\text{rank } A_{12}\} = n_U,
$$

where the maximum is taken over all symmetric partitions of the form

$$
A = \begin{bmatrix}
  * & A_{12} \\
  A_{21} & *
\end{bmatrix}.
$$

We next show how the scheme of Section 6.2 can be modified to reduce any symmetric matrix to a quasiseparable matrix. Consider again the $4 \times 4$ example

$$
A = \begin{bmatrix}
  a_{11} & a_{21} & a_{31} & a_{41} \\
  a_{21} & a_{22} & a_{32} & a_{42} \\
  a_{31} & a_{32} & a_{33} & a_{43} \\
  a_{41} & a_{42} & a_{43} & a_{44}
\end{bmatrix}.
$$

As opposed to (6.1), let us choose a matrix $P_2$ such that

$$
P_2 \begin{bmatrix}
  a_{31} \\
  a_{41}
\end{bmatrix} - \begin{bmatrix}
  a_{32} \\
  a_{42}
\end{bmatrix} = \begin{bmatrix}
  * \\
  0
\end{bmatrix},
$$

and hence

$$
P_2 \begin{bmatrix}
  a_{31} \\
  a_{41}
\end{bmatrix} = \begin{bmatrix}
  * \\
  \tilde{a}
\end{bmatrix}, \quad P_2 \begin{bmatrix}
  a_{32} \\
  a_{42}
\end{bmatrix} = \begin{bmatrix}
  \tilde{a} \\
  *
\end{bmatrix};
$$

that is, the two products differ only in their first entry. Then, similar to above, we have

$$
\begin{bmatrix}
  I_2 & 0 \\
  0 & P_2
\end{bmatrix}
\begin{bmatrix}
  a_{11} & a_{21} & a_{31} & a_{41} \\
  a_{21} & a_{22} & a_{32} & a_{42} \\
  a_{31} & a_{32} & a_{33} & a_{43} \\
  a_{41} & a_{42} & a_{43} & a_{44}
\end{bmatrix}
\begin{bmatrix}
  I_2 & 0 \\
  0 & P_2
\end{bmatrix}
= \begin{bmatrix}
  a_{11} & a_{21} & \tilde{a}^T & * \\
  a_{21} & a_{22} & \tilde{a}^T & * \\
  \tilde{a} & \tilde{a} & 0 & P_2A_1P_2
\end{bmatrix},
$$

(2, 2)-quasiseparable matrix

(1, 1)-out-of-band quasiseparable matrix

(6.2)

The above reduction step creates a dependence relation in the first and second rows, above the diagonal, and similarly in the first and second columns below it. In other words, the top-right and lower-left blocks of the first symmetric partition are low rank. Continuing in this fashion does not destroy this dependence, and results in a quasiseparable matrix. Two remarks are due.

**Remark 6.2.** We emphasize at this point that the classical reduction to tridiagonal is but a special case of the reduction described herein. Preceding what is described in this subsection with a Householder reflection $P_1$ of (6.1) (i.e., the one zeroing out the first column below the first two elements), the algorithm reduces precisely to tridiagonal form. Thus, we have described a family of reductions (parameterized by the choice of $P_1$, which can be arbitrary), one of which is the classical reduction to tridiagonal form.
Remark 6.3. Secondly, depending on partitioning the matrix in (6.2) one obtains off-diagonal blocks of different ranks. Specifically, the highlighted off-diagonal blocks of the left matrix in (6.2) clearly have rank at most two. On the other side, with a different partitioning, the matrix on the right has highlighted blocks of rank one. As we shall see in a moment, it is attractive to manipulate with blocks of lower ranks which motivates the definition given next.

Definition 6.4 (Out-of-Band Quasiseparable Matrices). A matrix $A$ is called out-of-band $(n_L, n_U)$-quasiseparable (with a bandwidth $(2k - 1)$) if \( \max(\text{rank} A_{31}) = n_L \) and \( \max(\text{rank} A_{13}) = n_U \), where the maximum is taken over all symmetric partitions of the form

$$A = \begin{bmatrix} * & * & A_{13} \\ * & A_{22} & * \\ A_{31} & * & * \end{bmatrix},$$

with any \( k \times k A_{22} \).

Basically, a matrix is quasiseparable if the blocks above/below the main diagonal in symmetric partitions are low rank. A matrix is off-band quasiseparable if the blocks above the first superdiagonal/below the first subdiagonal in symmetric partitions are low rank, as illustrated next.

![Illustration depicting which blocks have low rank in each structure](image)

Remark 6.5. The class of out-of-band quasiseparable matrices is more general (and it is more efficient in capturing the structure) than the class of band-plus-quasiseparable matrices. Indeed, consider a \((2m - 1)\)-band matrix (just as an example). It can be immediately seen that it is \((2m - 3)\)-band plus order \(m\) quasiseparable. At the same time it is clearly \((2m - 3)\)-band plus order 1 out-of-band quasiseparable, i.e., the order of quasiseparability drops from \(m\) to 1 leading to savings in storage and in reducing the cost.

6.4. QR iterations

Finally, the symmetric out-of-band quasiseparable structure is preserved under QR iterations. To show this, assume for simplicity that we start with an invertible matrix \(A^{(1)}\) and compute its QR factorization

$$A^{(1)} = \begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & A_{13}^{(1)} \\ A_{21}^{(1)} & A_{22}^{(1)} & A_{23}^{(1)} \\ A_{31}^{(1)} & A_{32}^{(1)} & A_{33}^{(1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(1)} & Q_{12}^{(1)} & Q_{13}^{(1)} \\ Q_{21}^{(1)} & Q_{22}^{(1)} & Q_{23}^{(1)} \\ Q_{31}^{(1)} & Q_{32}^{(1)} & Q_{33}^{(1)} \end{bmatrix} \cdot \begin{bmatrix} R_{11}^{(1)} & R_{12}^{(1)} & R_{13}^{(1)} \\ 0 & R_{22}^{(1)} & R_{23}^{(1)} \\ 0 & 0 & R_{33}^{(1)} \end{bmatrix}.$$

The next iterant, \(A^{(2)}\) is obtained via

$$A^{(2)} = \begin{bmatrix} A_{11}^{(2)} & A_{12}^{(2)} & A_{13}^{(2)} \\ A_{21}^{(2)} & A_{22}^{(2)} & A_{23}^{(2)} \\ A_{31}^{(2)} & A_{32}^{(2)} & A_{33}^{(2)} \end{bmatrix} = \begin{bmatrix} R_{11}^{(1)} & R_{12}^{(1)} & R_{13}^{(1)} \\ 0 & R_{22}^{(1)} & R_{23}^{(1)} \\ 0 & 0 & R_{33}^{(1)} \end{bmatrix} \cdot \begin{bmatrix} Q_{11}^{(1)} & Q_{12}^{(1)} & Q_{13}^{(1)} \\ Q_{21}^{(1)} & Q_{22}^{(1)} & Q_{23}^{(1)} \\ Q_{31}^{(1)} & Q_{32}^{(1)} & Q_{33}^{(1)} \end{bmatrix},$$

so that \(\text{rank} A_{31}^{(1)} = \text{rank} Q_{31}^{(1)} R_{11}^{(1)} = \text{rank} R_{33}^{(1)} Q_{31}^{(1)} = \text{rank} A_{31}^{(2)}\). The fact that \(\text{rank} A_{13}^{(1)} = \text{rank} A_{13}^{(2)}\) follows from symmetry.

We refer to [6, 1] for the details of the QR iteration algorithm for matrices with out-of-band quasiseparable structure, and describe here its basic steps.
1. We start with a \((r, r)\)-out-of-band quasiseparable matrix with the bandwidth \(m\) given by its generators. We first compute two unitary matrices \(V\) and \(U\) and an upper triangular matrix \(R\) whose out-of-band orders of quasiseparability are indicated below matrices, and the corresponding bandwidth is written on the top:

\[
\begin{align*}
\text{bandwidth} &= (2m - 1) \\
\widetilde{A}^{(1)}_{(r, r)} - \sigma I &= V_{(r, *)} \cdot U_{(\ast, 2r)} \cdot \widetilde{R}_{(0, 2r)} \\
\text{bandwidth} &= (4m - 2)
\end{align*}
\]

Here \(\sigma\) is a standard shift that can be chosen to speed up the convergence of the QR method.

2. Then we compute the generators of \(Q\) which is a product of \(V\) and \(U\):

\[
\begin{align*}
\widetilde{A}^{(1)}_{(r, r)} - \sigma I &= \widetilde{Q}_{(r, 2r)} \cdot \widetilde{R}_{(0, 2r)} \\
\text{bandwidth} &= (4m - 2)
\end{align*}
\]

3. Finally we compute the generators of the next iterant \(A^{(2)}\):

\[
\begin{align*}
\text{bandwidth} &= (2m - 1) \\
\widetilde{A}^{(2)}_{(r, r)} &= \widetilde{R}_{(0, 2r)} \cdot \widetilde{Q}_{(r, 2r)} + \sigma I
\end{align*}
\]

Since the algorithm avoids computations with matrix entries and manipulates only the generators, this is an efficient algorithm using only linear storage.

As with the reduction described earlier in this section, the previously considered classes of tridiagonal, banded, and semiseparable are special cases, and what is described is a generalization. As we have seen it above, these unifying approaches are typical of what is offered by working with the superclass of quasiseparable matrices.

### 7. Eigenvalue problems: Divide and conquer algorithms

Another (now also standard) method for computing eigenvalues replaces the second step of Section 6 with two different steps, based on the divide and conquer strategy \[15\]. This strategy is based on (i) the divide step, and (ii) the conquer step.

#### 7.1. Tridiagonal matrices and divide and conquer algorithms

The divide step consists of transforming a single problem into two smaller problems with the same structure of size roughly half of the original. This is done recursively until several small problems, say of size 2, can be solved very simply. The conquer step consists of "stitching together" all of these solutions to small problems. That is, one needs to take the solution of the two problems of half the size, and use them to compute the solution of the larger eigenvalue problem from which the smaller problems were "divided". An overview of divide and conquer methods was given in Section 2.

#### 7.2. Divide and conquer algorithms. Previous work

This idea was carried over from tridiagonal matrices to several other special structures, and the results of this work are given in Table 8.

In a recent paper \[1\], one can find an algorithm that is a generalization of all of this previous work listed in Table 8, as we consider the superclass of quasiseparable matrices. The input of our algorithm is a linear array of generators, and all known algorithms can be obtained via using special generators listed in Table 3.

#### 7.3. Generators of arbitrary order, non-Hessenberg quasiseparable matrices

We start with an analogue of Theorem 2.19, which provides generators of a \((n_L, n_U)\)-quasiseparable matrix in Definition 6.1.
Let $A$ be an $n \times n$ matrix. Then $A$ is an $(n_l, n_u)$-quasiseparable matrix if and only if there exists a set 
\{ $p_i, q_i, d_i, g_i, b_i, h_i$\} for $i = 1, \ldots, n - 1, j = 2, \ldots, n, k = 2, \ldots, n - 1,$ and $l = 1, \ldots, n,$ such that

\[ A = \begin{pmatrix} d_1 & g_1 b_1 h_1 \\ \vdots & \ddots & \ddots & \vdots \\ p_1 a_1 q_1 & & & d_n \\ & & & \\ & & & \\ \end{pmatrix} \]

The generators of the matrix $A$ are matrices of sizes

<table>
<thead>
<tr>
<th>sizes</th>
<th>$p_k$</th>
<th>$a_k$</th>
<th>$q_k$</th>
<th>$d_k$</th>
<th>$g_k$</th>
<th>$b_k$</th>
<th>$h_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>range</td>
<td>$1 \times r_{k-1}^l$</td>
<td>$r_k^l \times r_{k-1}^l$</td>
<td>$r_k^l \times 1$</td>
<td>$1 \times r_k^l$</td>
<td>$r_k^l \times r_k^l$</td>
<td>$r_k^l \times 1$</td>
<td></td>
</tr>
<tr>
<td>$k \in [2, n]$</td>
<td>$k \in [2, n-1]$</td>
<td>$k \in [1, n-1]$</td>
<td>$k \in [1, n]$</td>
<td>$k \in [1, n-1]$</td>
<td>$k \in [2, n-1]$</td>
<td>$k \in [2, n]$</td>
<td></td>
</tr>
</tbody>
</table>

where

\[ \max r_k^l = n_l \quad \max r_k^u = n_u \]

and

\[ a_{ij}^x = \begin{cases} a_{i-1} \cdots a_{j+1} & \text{for } i > j + 1 \\ 1 & \text{for } i = j + 1 \end{cases}, \quad b_{ij}^x = \begin{cases} b_{i+1} \cdots b_{j-1} & \text{for } j > i + 1 \\ 1 & \text{for } j = i + 1 \end{cases}. \]

Furthermore, using the generators of the previous theorem as well as the notations

\[ P_{m+1} = \text{col} \left( p_k a_{k,m}^x \right)_{k=m+1} = \begin{bmatrix} p_{m+1} \\ p_{m+2} a_{m+1} \\ p_{m+3} a_{m+2} a_{m+1} \\ \vdots \\ p_n a_{n-1} \cdots a_{m+2} a_{m+1} \end{bmatrix}, \]  

(7.1)

\[ Q_m = \text{row} \left( a_{m+1,k}^x q_k \right)_{k=1} = \begin{bmatrix} a_m \cdots a_2 a_1 q_1 & a_m \cdots a_2 q_2 & \cdots & a_m q_{m-1} & q_m \end{bmatrix}^T, \]  

(7.2)

one can represent a Hermitian quasiseparable matrix $A$ as

\[ A = \begin{bmatrix} A_m & Q_m^* \rho_{m+1}^* \\ P_{m+1} Q_m & B_{m+1} \end{bmatrix}. \]  

(7.3)

### 7.4. New divide and conquer algorithm for arbitrary order quasiseparable matrices

Suppose that $A$ is a Hermitian, quasiseparable matrix. Thus, by using the notations in Section 7.3, we have representation (7.3) for any Hermitian quasiseparable matrix, and so

\[ A = \begin{bmatrix} A_m & Q_m^* P_{m+1}^* \\ P_{m+1} Q_m & B_{m+1} \end{bmatrix} = \begin{bmatrix} A_m' & 0 \\ 0 & B_{m+1}' \end{bmatrix} + \begin{bmatrix} Q_m^* \\ P_{m+1}^* \end{bmatrix} \begin{bmatrix} Q_m \\ P_{m+1} \end{bmatrix}, \]  

(7.4)

with\(^6\)

\[ A_m' = A_m - Q_m^* Q_m, \quad B_{m+1}' = B_{m+1} - P_{m+1} P_{m+1}^*. \]  

(7.5)

Notice that this is an expression of $A$ as a block diagonal matrix plus a small rank matrix (the actual rank of this small rank matrix depends on the order of quasiseparability).

In order to specify the divide step of a divide and conquer algorithm, we must show that once completed, the divide and conquer algorithm could be applied recursively to the partitions $A_m'$ and $B_{m+1}'$ of (7.5). Thus we show next that the algorithm is applicable to $A_m'$ and $B_{m+1}'$; that is, they are Hermitian and quasiseparable. In fact, a slight modification of the next theorem gives a more general result also valid for non-Hermitian matrices.

\(^6\) When making this calculation we are assuming that these multiplications are well-defined, which can be accomplished by padding the generators to appropriate size.
Theorem 7.2. Let $A$ be a Hermitian, order $(k, k)$-quasiseparable matrix. Then the matrices

\[ A_m' = A_m - Q_m^* Q_m, \quad B_{m+1}' = B_{m+1} - P_{m+1}' P_{m+1}' \]

as defined above are Hermitian matrices that are quasiseparable of order at most $(k, k)$.

Proof. We give the proof for $A_m'$, the second part is analogous. Without loss of generality, we assume that $r_m' = r_m^k = k$ (see Theorem 7.1), which can easily be accomplished by padding given generators with zeros as needed. With this assumption, the product $Q_m^* Q_m$ is well defined and of the size $m \times m$ as required.

For $i > j$ (in the strictly lower triangular part), the definitions of $A$ and $Q_m$ yield the $(i, j)$-th entry of $A_m'$ to be

\[ A_m'(i, j) = A_m(i, j) - Q_m^*(i)Q_m(j) = p_i a_i^q q_i - q_i^*(a^*)_{m+1, 1}^m q_i \]

and from the equality

\[ a_{m+1, i}^\infty = a_m a_{m-1} \cdots a_{i+1} a_{i+1} \cdots a_{j-1} = a_{m+1, i}^\infty \]

this becomes

\[ A_m'(i, j) = p_i a_i^q q_i - q_i^*(a^*)_{m+1, 1}^m a_{m+1, 1}^m a_{m+1, 1}^m q_i = (p_i - q_i^*(a^*)_{m+1, 1}^m a_{m+1, 1}^m a_{m+1, 1}^m) a_{m+1, 1}^m q_i \tag{7.6} \]

giving the desired quasiseparable structure in the strictly lower triangular portion. From (7.6), we see that the sizes of the resulting lower generators of $A_m'$ do not exceed those of $A_m$, and hence $A_m'$ is lower quasiseparable of order at most $k$.

For the main diagonal, we similarly obtain

\[ A_m'(i, i) = d_i - g_i a_{m+1, 1}^m q_i \]

and for the strictly upper triangular part where $j > i$, we note that since both $A_m$ and $Q_m^* Q_m$ are Hermitian, their difference is as well, and the upper part is hence also quasiseparable of the same order. This completes the proof. \( \Box \)

From this theorem it is clear that the quasiseparable structure of $A$ is inherited by the matrices $A_m'$ and $B_{m+1}'$, and the Hermitian property is as well. This specifies the divide step of the algorithm.

Thus the eigenproblem is decomposed into two smaller eigenproblems, and by using a small-rank update they can be solved using the common conquer step of [15].

8. Summary

In this paper we used the relatively new concept of quasiseparable matrices to generalize the algorithms of Traub, Björck–Pereyra and Cuppen. These generalizations were based on the interplay between polynomial sequence with sparse recurrence relations and classes of structured matrices. This is as opposed to previous approaches to obtaining algorithms for e.g. the Szegő case, which yielded not generalizations but alternate algorithms valid only for that case. Our approach using the quasiseparable structure is unifying; special choices of generators in our algorithm correspond to special cases of quasiseparable matrices, and in this sense our method generalizes all previous works.

Acknowledgements

The authors would like to thank Dario A. Bini, Victor Y. Pan, and the anonymous referee for a number of suggestions that allowed us to improve the exposition.

References
