

Computations with quasiseparable polynomials and matrices

T.Bella*, Y.Eidelman[†], I.Gohberg[†], V.Olshevsky*,

Abstract

In this paper we survey several recent results that highlight an interplay between a relatively new class of *quasiseparable* matrices and 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 Szegő polynomials, resp. 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. Recently, it has been of interest to derive variations of these algorithms that are valid for the class of Szegő polynomials. However, such new algorithms tend to be valid only for the Szegő polynomials; they are analogues and not generalizations of the original algorithms.

Herein we survey several results recently obtained for the “superclass” of *quasiseparable* matrices, which includes both Jacobi and unitary Hessenberg matrices as special cases. Hence the interplay between quasiseparable matrices and their polynomial systems (which contain both real orthogonal and Szegő polynomials) allows one to obtain true generalizations of several algorithms. The specific algorithms and topics that we discuss are the Björck–Pereyra algorithm, the Traub algorithm, certain new digital filter structures, as well as QR and divide and conquer eigenvalue algorithms.

Keywords. Semiseparable matrices, quasiseparable matrices, orthogonal polynomials, Szegő polynomials, recurrence relations, QR iteration, divide and conquer eigenvalue algorithms, signal flow graphs, digital filters, Björck–Pereyra algorithm, Traub algorithm.

1 Introduction

An interplay between polynomials and structured matrices is a well-studied topic, see, e.g., [47, 43, 44, 45] and many references therein. In the context of polynomial computations, typically matrices with Toeplitz, Hankel, Vandemonde, and related structures were of interest.

Recently, a rather different class of *quasiseparable matrices* has been receiving a lot of attention¹. 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, 1.2, 1.3) one of the differences between classical classes of structured matrices and the new one.

*Department of Mathematics, University of Connecticut, Storrs CT 06269-3009, USA. Email: bella@math.uconn.edu

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

¹Quasiseparable 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 [18, 19, 9, 22, 16, 55] 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 polynomial systems.

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

$$\langle p(x), q(x) \rangle = \int_a^b p(x)q(x)w^2(x)dx, \quad \text{where } w^2(x) \text{ is some weight function,} \quad (1.1)$$

and such polynomials are classical [49, 29]. They arise in a variety of problems in scientific computing such as numerical integration/Gaussian quadrature [50], discrete sine/cosine transforms [32], systems and control [17], and solving differential equations [34].

Numerous applications in signal processing [46], system theory and control [35], inverse scattering [36] give rise to other orthogonal polynomials, namely the Szegő polynomials, which are polynomials orthogonal with respect to the inner product $\langle \cdot, \cdot \rangle$ defined via integration on the unit circle, i.e.,

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

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

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

Both of these moment matrices H and T are shift-invariant, (i.e., they have constant values along their antidiagonals and diagonals, resp.). This can be immediately deduced from the definitions of their corresponding inner product $\langle \cdot, \cdot \rangle$ above, along with the definition $m_{kj} = \langle x^k, x^j \rangle$ for the *moments* (i.e., for the entries of the corresponding moment matrix $M = [m_{kj}]$). Indeed, in the real line case it follows from (1.1) that $m_{kj} = \int_a^b x^{k+j}w^2(x)dx$ depends on the sum of the row and column indices. Hence the matrix M has a Hankel structure². This shift-invariant structure of H and T clearly means that these two square arrays are *structured*, i.e., they are defined by only $\mathcal{O}(n)$ 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 Szegő polynomials, namely tridiagonal and unitary Hessenberg matrices, respectively, displayed

²The structure of the Toeplitz matrix is deduced similarly from (1.2) and the fact that on the unit circle we have $\overline{x^j} = \overline{e^{ij\theta}} = e^{-ij\theta} = x^{-j}$, so that each moment m_{kj} now depends only on the difference of indices, which yields the Toeplitz structure.

next.

$$T_n = \begin{bmatrix} \delta_1 & \gamma_2 & 0 & \cdots & 0 \\ \gamma_2 & \delta_2 & \gamma_3 & \ddots & \vdots \\ 0 & \gamma_3 & \delta_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \gamma_n \\ 0 & \cdots & 0 & \gamma_n & \delta_n \end{bmatrix}, \quad U_n = \begin{bmatrix} -\rho_0^* \rho_1 & -\rho_0^* \mu_1 \rho_2 & -\rho_0^* \mu_{1,2} \rho_3 & \cdots & -\rho_0^* \mu_{1,n-1} \rho_n \\ \mu_1 & -\rho_1^* \rho_2 & -\rho_1^* \mu_2 \rho_3 & \cdots & -\rho_1^* \mu_{2,n-1} \rho_n \\ 0 & \mu_2 & -\rho_2^* \rho_3 & \cdots & -\rho_2^* \mu_{3,n-1} \rho_n \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mu_{n-1} & -\rho_{n-1}^* \rho_n \end{bmatrix}, \quad (1.4)$$

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

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

that real orthogonal polynomials $\{r_k(x)\}$ are known to satisfy [50]. 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 Szegő polynomials. This justifies the nomenclature *recurrence matrices* used for T_n and U_n .

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 [37, 31, 13, 33, 45, 42, 41] 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 [37], 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 $\mathcal{O}(n)$ 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 [18, 19]. 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*³. The point is that historically, algorithm development is

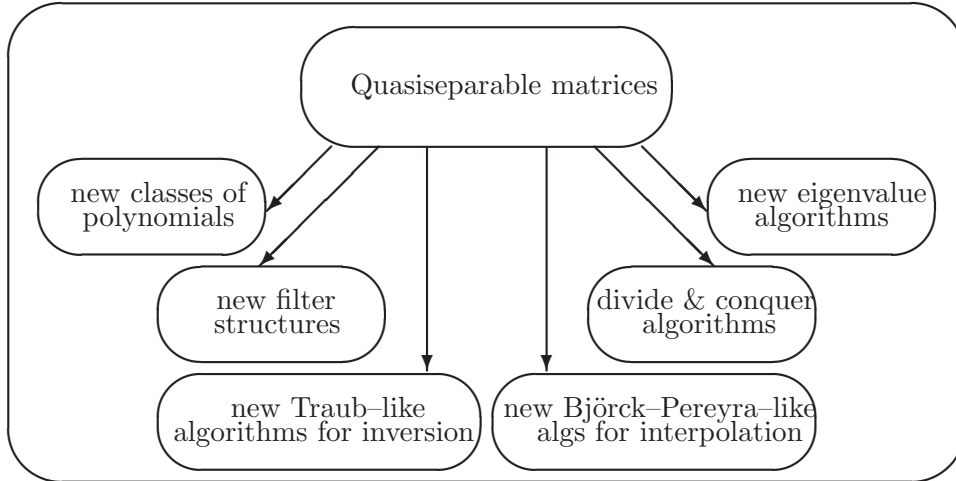
³Polynomials corresponding to semiseparable matrices (which is, along with tridiagonal and unitary Hessenberg matrices, another subclass of quasiseparable matrices) were recently studied in [23].

Table 1: Polynomial families and their recurrence and moment matrices

Polynomials	Moment matrices	Recurrence matrices
real-orthogonal	Hankel	tridiagonal
Szegö	Toeplitz	unitary Hessenberg
↓		
Generalizations	matrices with displacement structure	Quasiseparable

more advanced for real orthogonal polynomials. Recently, several important algorithms originally derived for real orthogonal polynomials have subsequently 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).

Considering 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.



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 some system 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). In particular, 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 a Divide and Conquer method, due to Cuppen [14, 12]. The basic idea of a divide and conquer method is that **(i)** a problem of size n can be subdivided into two problems with the same structure of size about $n/2$ (the divide step), and **(ii)** if we know the solution to the two smaller problems of size $n/2$, 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 solved by traditional methods, and a solution 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

$$T = \left[\begin{array}{ccc|ccc} \star & \star & 0 & 0 & 0 & 0 \\ \star & \star & \star & 0 & 0 & 0 \\ 0 & \star & \star & \star & 0 & 0 \\ 0 & 0 & \star & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & 0 & \star & \star \end{array} \right] = \left[\begin{array}{ccc|ccc} \star & \star & 0 & 0 & 0 & 0 \\ \star & \star & \star & 0 & 0 & 0 \\ 0 & \star & \star & 0 & 0 & 0 \\ 0 & 0 & 0 & \star & \star & 0 \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & 0 & \star & \star \end{array} \right] + A = \left[\begin{array}{c|c} T_1 & 0 \\ \hline 0 & T_2 \end{array} \right] + A \quad (1.6)$$

where $\text{rank}(A) = 1$. This observation leads to the divide step, and together with the conquer step, a method for obtaining the eigendecomposition of T given A and the eigendecompositions of T_1 and T_2 together allow calculation of the eigendecomposition of a tridiagonal matrix, and thus the roots of real orthogonal polynomials.

“Carried over” algorithm: Szegő polynomials. It was noticed by Gragg and Reichel in [28] that a similar algorithm was possible for computing the roots of Szegő polynomials. Szegő 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 Szegő polynomial, and thus we seek an eigendecomposition of U_n . The Divide and Conquer algorithm of [28] 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 $G_k(\rho_k)$,

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

see, e.g., [24]. 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)

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

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

That is, 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 Szegő 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

$$A = \left[\begin{array}{c|c} * & A_{12} \\ \hline A_{21} & * \end{array} \right].$$

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

$$A_{21} = \begin{bmatrix} 0 & \cdots & 0 & \star \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix}$$

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

$$A_{12} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \star & 0 & \cdots & 0 \end{bmatrix}$$

and

$$A_{12} = \begin{bmatrix} -\rho_k \mu_{k-1} \cdots \mu_3 \mu_2 \mu_1 \rho_0^* & -\rho_{k-1} \mu_{k-2} \cdots \mu_3 \mu_2 \mu_1 \rho_0^* & \cdots & -\rho_n \mu_{n-1} \cdots \mu_3 \mu_2 \mu_1 \rho_0^* \\ -\rho_k \mu_{k-1} \cdots \mu_3 \mu_2 \rho_1^* & -\rho_{k-1} \mu_{k-2} \cdots \mu_3 \mu_2 \rho_1^* & \cdots & -\rho_n \mu_{n-1} \cdots \mu_3 \mu_2 \rho_1^* \\ -\rho_k \mu_{k-1} \cdots \mu_3 \rho_2^* & -\rho_{k-1} \mu_{k-2} \cdots \mu_3 \rho_2^* & \cdots & -\rho_n \mu_{n-1} \cdots \mu_3 \rho_2^* \end{bmatrix}, \quad (1.8)$$

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

- Szegő polynomials lead to the so-called lattice digital filter structures. We describe more general semiseparable and quasiseparable filter structures. (Section 3).
- Björck-Pereyra algorithms were carried over from three-term-Vandermonde matrices to Szegő-Vandermonde matrices ([2]). We generalize these algorithms to the quasiseparable case. (Section 4).
- Traub algorithms were carried over from three-term-Vandermonde matrices to Szegő-Vandermonde matrices ([39, 40]). 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 systems of polynomials. Let \mathcal{H} be the set of all upper strongly Hessenberg matrices (“strongly” means that $a_{i+1,i} \neq 0$ for $i = 1, \dots, n-1$ and $a_{i,j} = 0$ for $i > j+1$), and \mathcal{P} be the set of all polynomial systems $\{r_k(x)\}$ 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

$$f(A) = P, \quad P = \{r_k\}_{k=0}^n, \quad r_k(x) = \frac{1}{a_{2,1}a_{3,2} \cdots a_{k,k-1}} \det(xI - A)_{k \times k}; \quad (2.1)$$

that is, associate with each upper strongly Hessenberg matrix H the polynomial system consisting of the (scaled) characteristic polynomials of the principal submatrices of H . It is clear that this provides a mapping from matrices to polynomial systems, i.e. $f : \mathcal{H} \rightarrow \mathcal{P}$. 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].

Proposition 2.1. *The function $f : \mathcal{H} \rightarrow \mathcal{P}$ 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] 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 Szegő polynomials (to be formally defined below by (2.4)) is related via (2.1) to the class of (almost⁴) unitary Hessenberg matrices; that is, those of the form

$$H = \begin{bmatrix} -\rho_0^* \rho_1 & -\rho_0^* \mu_1 \rho_2 & -\rho_0^* \mu_1 \mu_2 \rho_3 & \cdots & -\rho_0^* \mu_1 \mu_2 \mu_3 \cdots \mu_{n-1} \rho_n \\ \mu_1 & -\rho_1^* \rho_2 & -\rho_1^* \mu_2 \rho_3 & \cdots & -\rho_1^* \mu_2 \mu_3 \cdots \mu_{n-1} \rho_n \\ 0 & \mu_2 & -\rho_2^* \rho_3 & \cdots & -\rho_2^* \mu_3 \cdots \mu_{n-1} \rho_n \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mu_{n-1} & -\rho_{n-1}^* \rho_n \end{bmatrix} \quad (2.2)$$

$$\text{with } \rho_0 = -1, \quad |\rho_k| < 1, \quad k = 1, \dots, n-1, \quad |\rho_n| \leq 1; \quad \mu_k = \begin{cases} \sqrt{1 - |\rho_k|^2} & |\rho_k| < 1 \\ 1 & |\rho_k| = 1 \end{cases}.$$

⁴A matrix H is called a(n) almost unitary Hessenberg matrix provided $H = UD$ for a unitary Hessenberg matrix U and diagonal matrix $D = \text{diag}\{1, \dots, 1, \rho_n\}$. It is unitary except for possibly a scaling of the last column.

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 Szegő polynomials. Another class of polynomials for which results valid for real orthogonal polynomials have been extended (see [23]) are 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 $\text{triu}(A_U, 1)$).*

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, Szegő, 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 ($a_{i+1,i} \neq 0$ for $i = 1, \dots, n-1$ and $a_{i,j} = 0$ for $i > j+1$), and (ii) $\max(\text{rank} A_{12}) = 1$ where the maximum is taken over all symmetric partitions of the form*

$$A = \left[\begin{array}{c|c} * & A_{12} \\ \hline * & * \end{array} \right].$$

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.

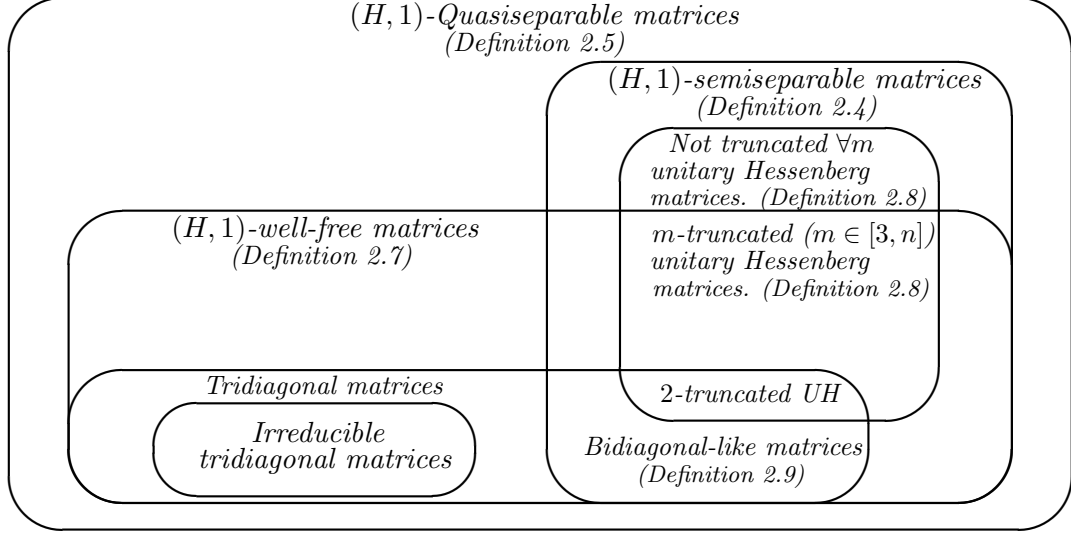
Conversely, however, consider a tridiagonal matrix. Any symmetric partition A_{12} of a tridiagonal matrix as in Definition 2.5 will have at most one nonzero entry, and thus be rank one. Thus, tridiagonal matrices are $(H, 1)$ -quasiseparable. However, they are not necessarily $(H, 1)$ -semiseparable. 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

$$\left[\begin{array}{cccccc} ? & \star & 0 & \dots & 0 \\ ? & ? & \star & \ddots & \vdots \\ ? & ? & ? & \ddots & 0 \\ \vdots & & \ddots & \ddots & \star \\ ? & ? & \dots & ? & ? \end{array} \right],$$

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] 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:*

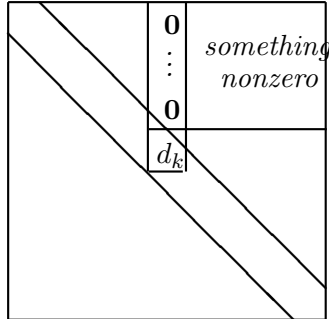


The proof of this theorem can be found in [4], and we next present the details of the subclasses introduced in the figure.

Definition 2.7 ($(H, 1)$ -well-free matrices).

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

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 $k = 2, \dots, n-1$ 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 $\rho_2 \neq 0, \dots, \rho_m \neq 0, \quad \rho_{m+1} = \dots = \rho_n = 0$.*

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 recurrent 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 Szegő polynomials are known to be related via (2.1) to tridiagonal, and unitary Hessenberg matrices, resp. These two facts imply that these polynomial classes satisfy sparse (e.g., three-term or two-term) recurrence relations. Moreover, the sparseness of these recurrence 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 Szegő polynomials.

Definition 2.10 (Real orthogonal polynomials and three-term recurrence relations). *A system of polynomials is real orthogonal if it satisfies the three-term recurrence relations*

$$r_k(x) = (\alpha_k x - \delta_k) r_{k-1}(x) - \gamma_k \cdot r_{k-2}(x), \quad \alpha_k \neq 0, \gamma_k > 0 \quad (2.3)$$

Definition 2.11 (Szegő polynomials and two-term recurrence relations). *A system of polynomials are Szegő polynomials $\{\phi_k^\# \}$ if they satisfy the two-term recurrence relations (with some auxiliary polynomials $\{\phi_k\}$)*

$$\begin{bmatrix} \phi_k(x) \\ \phi_k^\#(x) \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} 1 & -\rho_k^* \\ -\rho_k & 1 \end{bmatrix} \begin{bmatrix} \phi_{k-1}(x) \\ x\phi_{k-1}^\#(x) \end{bmatrix}, \quad (2.4)$$

where the reflection coefficients ρ_k and complementary parameters μ_k satisfy

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

Definition 2.12 ($(H, 1)$ -semiseparable polynomials and Szegő-type two-term recurrence relations). *A system of polynomials $\{r_k(x)\}$ is called $(H, 1)$ -semiseparable if it satisfies*

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ (\delta_k x + \theta_k) r_{k-1}(x) \end{bmatrix}. \quad (2.5)$$

The relations (2.5) are called Szegő-type two-term recurrence relations since they generalize the classical Szegő relations (2.4).

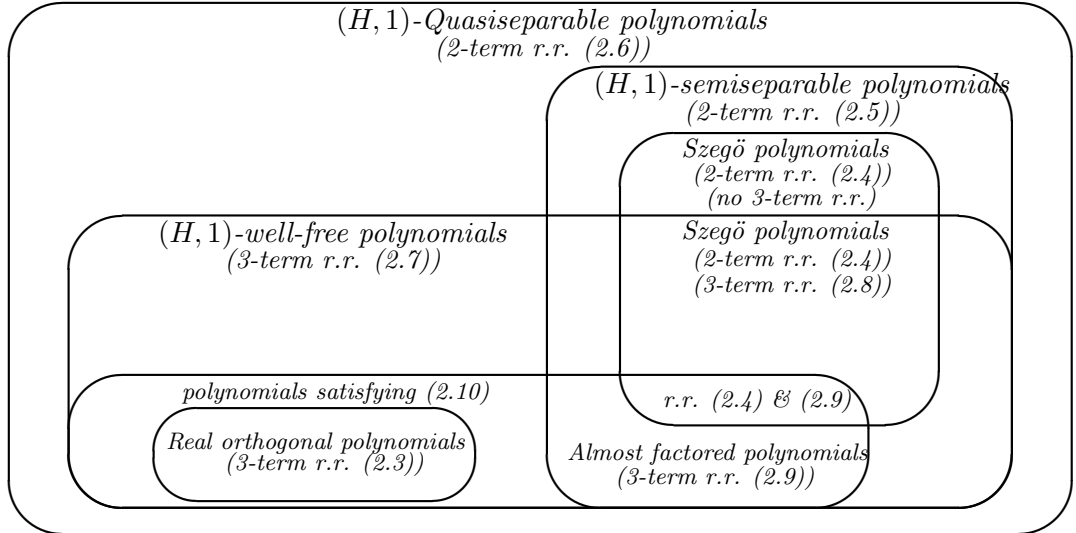
Definition 2.13 ($(H, 1)$ -quasiseparable polynomials and [EGO05]-type two-term recurrence relations). A system of polynomials $\{r_k(x)\}$ is called $(H, 1)$ -quasiseparable if it satisfies the following two-term recurrence relations

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & \delta_k x + \theta_k \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ r_{k-1}(x) \end{bmatrix}. \quad (2.6)$$

The relations (2.6) are called [EGO05]-type two-term recurrence relations since they are a special case of another formula given [21].

In order to provide a complete picture of the relationship between real orthogonal polynomials, Szegő 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.

Theorem 2.14. [4] 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], and it is based on the following fact that seems to be of interest by itself. One can observe that the shape and the configuration of ovals are identical in the two figures of Theorems 2.6 and 2.14. This is a reflection of the already mentioned fact, proven in [4], that for each pair of identical ovals we have the following. The polynomials described by any given oval in the figure of Theorem 2.14 are related via (2.1) to the matrices described by the corresponding oval in the figure of Theorem 2.6. More details on this will be given in Sections 2.4.2, 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 system of polynomials is called $(H, 1)$ -well-free if it satisfies the general three-term recurrence relations

$$r_k(x) = (\alpha_k x - \delta_k) \cdot r_{k-1}(x) - (\beta_k x + \gamma_k) \cdot r_{k-2}(x). \quad (2.7)$$

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

$$\phi_k^\#(x) = \begin{cases} \frac{1}{\mu_0} & k = 0 \\ \frac{1}{\mu_1}(x \cdot \phi_0^\#(x) + \rho_1 \rho_0^* \cdot \phi_0^\#(x)) & k = 1 \\ \frac{1}{\mu_2}x\phi_1^\#(x) - \frac{\rho_2 \mu_1}{\mu_2} \phi_0^\#(x) & k = 2, \rho_1 = 0 \\ \left[\frac{1}{\mu_2} \cdot x + \frac{\rho_2}{\rho_1} \frac{1}{\mu_2} \right] \phi_1^\#(x) - \frac{\rho_2 \mu_1}{\rho_1 \mu_2} \cdot x \cdot \phi_0^\#(x) & k = 2, \rho_1 \neq 0 \\ \left[\frac{1}{\mu_k} \cdot x + \frac{\rho_k}{\rho_{k-1}} \frac{1}{\mu_k} \right] \phi_{k-1}^\#(x) - \frac{\rho_k}{\rho_{k-1}} \frac{\mu_{k-1}}{\mu_k} \cdot x \cdot \phi_{k-2}^\#(x) & 2 < k \leq m \\ x \cdot \phi_{k-1}^\#(x) & k > m \end{cases} \quad (2.8)$$

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

$$r_k(x) = \begin{cases} (\alpha_k x - \delta_k) \cdot r_{k-1}(x) & k \neq j \\ ((\alpha_{k-1} x - \delta_{k-1})(\alpha_k x - \delta_k) - \gamma_k) \cdot r_{k-2}(x) & k = j \end{cases} \quad (2.9)$$

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

$$r_k(x) = (\alpha_k x - \delta_k) r_{k-1}(x) - \gamma_k \cdot r_{k-2}(x), \quad \alpha_k \neq 0; \quad (2.10)$$

that is, the same recurrence relations as satisfied by real orthogonal polynomials, but without the restriction of $\gamma_k > 0$.

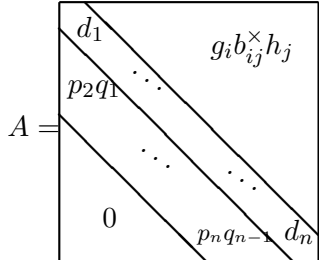
2.4 Sparse generators and an interplay between subclasses of $(H, 1)$ -quasiseparable matrices and $(H, 1)$ -quasiseparable polynomials

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 $\mathcal{O}(n^2)$ coefficients of the polynomials themselves, one can carry out computations with the $\mathcal{O}(n)$ 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 $\mathcal{O}(n)$ 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 $\{p_j, q_i, d_l, g_i, b_k, h_j\}$ for $i = 1, \dots, n-1$, $j = 2, \dots, n$, $k = 2, \dots, n-1$, and $l = 1, \dots, n$, such that*



$A =$

where $b_{ij}^\times = b_{i+1} \cdots b_{j-1}$ for $j > i + 1$
and $b_{ij}^\times = 1$ 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 ’s), this notation is consistent with the standard notation used in [21] for the more general (non-Hessenberg) case where both p ’s and q ’s are needed.

Definition 2.21. A set of elements $\{p_j, q_i, d_l, g_i, b_k, h_j\}$ as in the previous theorem for a matrix A are called **generators** of the matrix A . Note that generators of a given matrix are not unique, and this nonuniqueness extends nontrivially past the noted redundancy in p 's and q 's.

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

We next present two theorems from [4] 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 system of polynomials R corresponding to A satisfies the recurrence relations*

$$\begin{bmatrix} F_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{p_{k+1}q_k} \begin{bmatrix} q_k p_k b_k & -q_k g_k \\ p_k h_k & x - d_k \end{bmatrix} \begin{bmatrix} F_{k-1}(x) \\ r_{k-1}(x) \end{bmatrix}. \quad (2.12)$$

Theorem 2.23 ([EGO05]-type recurrence relation coefficients \Rightarrow quasiseparable generators). *Suppose that a system 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

$$\begin{aligned} p_k &= 1, & k &= 2, \dots, n, & q_k &= 1/\delta_k, & k &= 1, \dots, n-1, & d_k &= -\frac{\theta_k}{\delta_k}, & k &= 1, \dots, n \\ g_k &= \beta_k, & k &= 1, \dots, n-1, & b_k &= \alpha_k, & k &= 2, \dots, n-1, & h_k &= -\frac{\gamma_k}{\delta_k}, & k &= 2, \dots, n \end{aligned}$$

corresponds to R via (2.1).

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 following 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 providing 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 special restrictions for are given in Table 3. Note that a \star 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

Table 2: Characterizations of subclasses of $(H, 1)$ -quasiseparable matrices via related polynomials

Class of matrices	Class of polynomials
Irreducible tridiagonal matrices (1.4)	real orthogonal polynomials three-term (2.3)
Unitary Hessenberg matrices (2.2)	Szegő polynomials two-term (2.4)
$(H, 1)$ -semiseparable matrices Definition 2.4	$(H, 1)$ -semiseparable polynomials Szegő-type 2-term (2.5)
$(H, 1)$ -well-free matrices Definition 2.7	$(H, 1)$ -well-free polynomials general 3-term (2.7)
$(H, 1)$ -quasiseparable matrices Definition 2.5	$(H, 1)$ -quasiseparable polynomials [EGO05]-type 2-term (2.6)

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

Matrix class	p_j	q_i	d_l	g_i	b_k	h_j
Tridiagonal	1	$1/\alpha_i$	δ_l/α_l	$\gamma_{i+1}/\alpha_{i+1}$	0	1
Unitary Hessenberg	1	μ_i	$-\rho_{l-1}^* \rho_l$	$-\rho_{i-1}^* \mu_i$	μ_k	ρ_j
$(H, 1)$ -semiseparable	$\neq 0$	$\neq 0$	\star	\star	$\neq 0$	\star
$(H, 1)$ -well-free	$\neq 0$	$\neq 0$	\star	\star	\star	$\neq 0$
Bidiagonal-like	$\neq 0$	$\neq 0$	\star	$\neq 0$ at most once	0	$\neq 0$

$(H, 1)$ -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 these problems.

- Secondly, one needs to provide 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] for full 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 Section 2 were recently extended to the more general (H, m) -quasiseparable case in [8].

3 New quasiseparable digital filter structures

3.1 Filter structures and the Markel–Grey filter structure

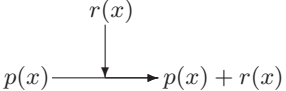
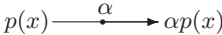
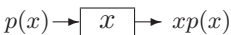
Complementing the algebraic descriptions of polynomials, particularly those determined by sparse recurrence relations, one can consider the corresponding signal flow graphs. Briefly, the goal is to

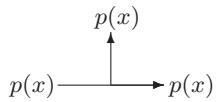
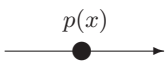
Table 4: Conversion formulas of the type “Recurrence relation coefficients \Rightarrow quasiseparable generators”

Polynomials	p_k	q_k	d_k	g_k	b_k	h_k
Monomials	1	1	0	0	0	1
Real orth. (2.3)	1	$1/\alpha_k$	δ_k/α_k	γ_k/α_k	0	1
Szegö (2.4)	1	μ_k	$-\rho_k \rho_{k-1}^*$	ρ_{k-1}^*	μ_{k-1}	$-\mu_{k-1} \rho_k$
Gen. 3-term (2.7)	1	$1/\alpha_k$	$\frac{\delta_k}{\alpha_k} + \frac{\beta_k}{\alpha_{k-1} \alpha_k}$	$\frac{d_k \beta_{k+1} + \gamma_{k+1}}{\alpha_{k+1}}$	$\frac{\beta_{k+1}}{\alpha_{k+1}}$	1
Szegö-type (2.5)	1	$1/\delta_k$	$-\frac{\theta_k + \gamma_k \beta_{k-1}}{\delta_k}$	β_{k-1}	$\alpha_{k-1} - \beta_{k-1} \gamma_{k-1}$	$-\frac{\gamma_k}{\delta_k} (\alpha_{k-1} - \beta_{k-1} \gamma_{k-1})$
[EGO05]-type (2.6)	1	$1/\delta_k$	$-\frac{\theta_k}{\delta_k}$	β_k	α_k	$-\frac{\gamma_k}{\delta_k}$

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.

Table 5: Building blocks of signal flow graphs

Adder	Gain	Delay
		
Implements polynomial addition.	Implements scalar multiplication.	Implements multiplication by x .

Splitter	Label
	
Allows a given signal to be used in multiple places.	Identifies the current signal (just for clarity, does not require an actual device).

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

$$p(x) = P_0 + P_1 x + P_2 x^2 + P_3 x^3 \quad (3.13)$$

by a signal flow graph is shown next in Figure 1.

An important result in signal processing is the *Markel–Grey* filter design, which realizes polynomials using the Szegö polynomials as a basis. The filter design uses the two-term recurrence relations (2.4), which gives the ladder structure shown in Figure 2 (page 17).

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

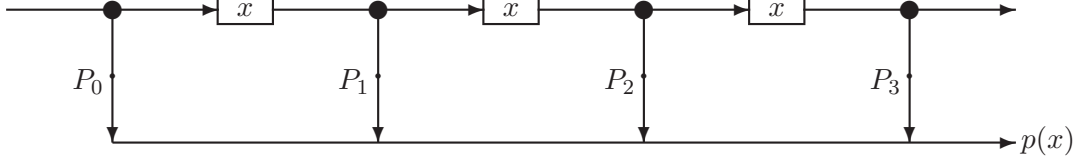


Figure 1: Implementation of the polynomial (3.13) expressed by a signal flow graph.

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 Figure 3, and the recurrence relations (2.6) lead to the *quasiseparable filter structure*, depicted in Figure 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 systems. 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 connections 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 system solver

4.1 Classical Björck–Pereyra algorithm and its extensions

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

$$V(x) = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}, \quad (4.1)$$

is classical, with applications in interpolation, coding theory, etc. It is known that Vandermonde matrices are extremely ill-conditioned [52], 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 $\mathcal{O}(n^3)$ 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 $\mathcal{O}(n)$ storage, and **(iii)** it is fast, using only $\mathcal{O}(n^2)$ flops⁵. The Björck–Pereyra algorithm is based on the following factorization of the inverse of a Vandermonde matrix:

$$V(x)^{-1} = U_1 \cdots U_{n-1} \cdot L_{n-1} \cdots L_1, \quad (4.2)$$

⁵It is easy to solve a Vandermonde system in $\mathcal{O}(n \log n)$ flops but such *superfast* algorithms are typically totally inaccurate already for 15×15 matrices.

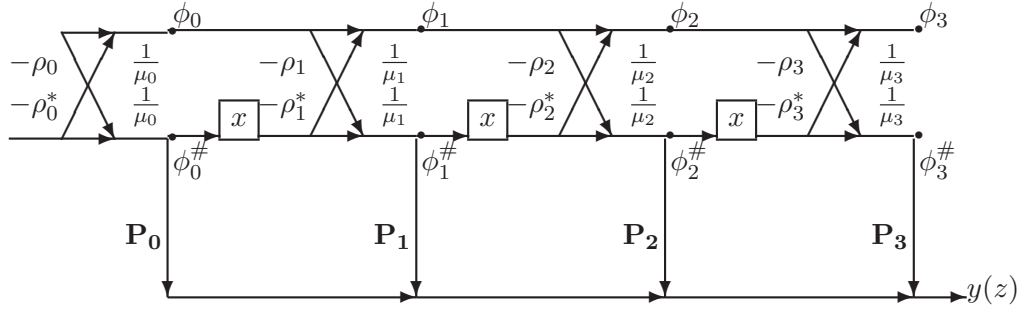


Figure 2: Markel-Grey filter structure: signal flow graph to realize the Szegő polynomials using two-term recurrence relations (2.4).

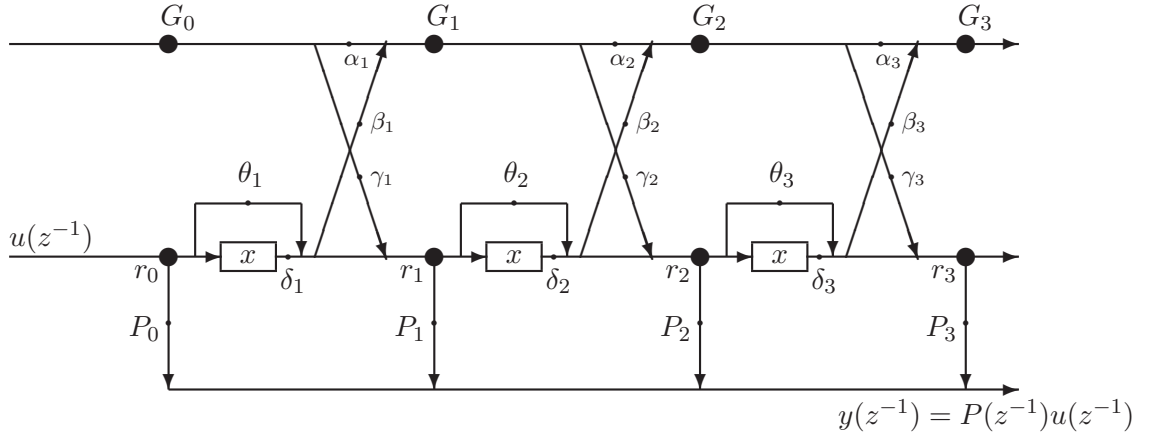


Figure 3: Semiseparable filter structure: Signal flow graph to realize polynomials R satisfying Szegő-type recurrence relations (2.5).

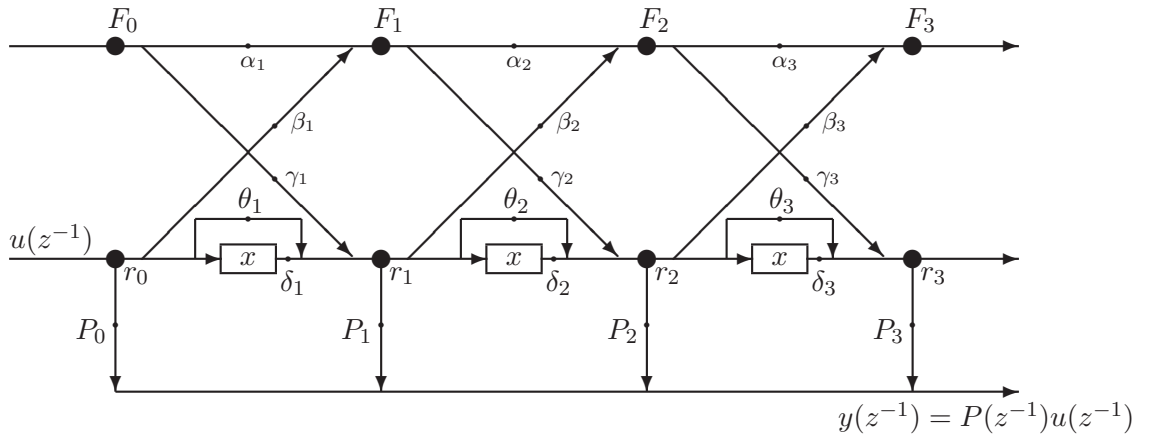


Figure 4: Quasiseparable filter structure: Signal flow graph for polynomials R using [EGO05]-type recurrence relations (2.6).

with certain **bidiagonal** 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 to be computed rather 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 $\mathcal{O}(n^2)$ operations are needed. Further, since the entries of U_k, L_k are readily defined by $\{x_i\}$, only $\mathcal{O}(n)$ storage is needed.

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

$$V_R(x) = \begin{bmatrix} r_0(x_1) & r_1(x_1) & \cdots & r_{n-1}(x_1) \\ \vdots & \vdots & & \vdots \\ r_0(x_n) & r_1(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix}, \quad (4.3)$$

namely to those specified in Table 6.

Table 6: Previous work. Fast $\mathcal{O}(n^2)$ polynomial–Vandermonde solvers.

Polynomial systems $R = \{r_k(x)\}$	Fast $\mathcal{O}(n^2)$ linear system solver
monomials	Björck–Pereyra [10]
Chebyshev polynomials	Reichel–Opfer [48]
Real orthogonal (three-term) polynomials	Higham [30]
Szegő polynomials	Bella–Eidelman–Gohberg–Koltracht–Olshevsky [2]

The algorithms of [48, 30, 2] are not generalizations of the original Björck–Pereyra [10] algorithm but analogs of it.

4.2 A true generalization. New Björck-Pereyra-type algorithm for quasiseparable-Vandermonde matrices

A new Björck-Pereyra-type algorithm, obtained recently in [3], 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 $V_R(x)$ be a polynomial–Vandermonde matrix of the form (4.3) for a system of $(H, 1)$ –quasiseparable polynomials R given by the generators $\{p_j, q_i, d_l, g_i, b_k, h_j\}$ of the corresponding $(H, 1)$ –quasiseparable matrix A in (2.11). Then*

$$V_R(x)^{-1} = U_1 \cdot \left[\begin{array}{c|c} I_1 & \\ \hline & U_2 \end{array} \right] \cdots \left[\begin{array}{c|c} I_{n-2} & \\ \hline & U_{n-1} \end{array} \right] \cdot \left[\begin{array}{c|c} I_{n-2} & \\ \hline & L_{n-1} \end{array} \right] \cdots \left[\begin{array}{c|c} I_1 & \\ \hline & L_2 \end{array} \right] \cdot L_1, \quad (4.4)$$

$$U_k = \left[\begin{array}{c|c} \frac{1}{\alpha_0} & \\ \hline 0 & A_{(n-k) \times (n-k)} - x_k I \\ \vdots & \\ 0 & \cdots 0 \frac{1}{\alpha_{n-k}} \end{array} \right], \quad L_k = \left[\begin{array}{c|c} 1 & \\ \hline \frac{1}{x_{k+1}-x_k} & \\ & \ddots \\ & \frac{1}{x_n-x_k} \end{array} \right] \left[\begin{array}{cccc} 1 & & & \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{array} \right], \quad (4.5)$$

where $A_{(n-k) \times (n-k)}$ is the leading $(n-k) \times (n-k)$ submatrix of A in (2.11), and $\alpha_0 = r_0(x)$ and for $k > 0$ the number α_k is the ratio of the leading coefficients of $r_k(x)$ and $r_{k-1}(x)$.

Remark 4.2. The overall cost for solving $V_R(x)a = f$ via (4.4) is $\mathcal{O}(n^2)$ operations, since matrices L_k, U_k can be multiplied by a vector in $\mathcal{O}(n)$ 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] for details.

At first glance the $\mathcal{O}(n^2)$ complexity may not seem optimal. However, even for the well-studied simplest Vandermonde matrices in (4.1), all algorithms with lower complexity are known to be quite inaccurate 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) are 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, solving problems with classical Vandermonde matrices in (4.1) is problematic due to their large condition numbers ([52]), 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 $\mathcal{O}(n^3)$ flops.

There is a fast Traub algorithm [51] that computes the entries of $V(x)^{-1}$ in only $\mathcal{O}(n^2)$ operations, but it is known to be inaccurate as well. Fortunately, it was observed in [27] 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 next.

Table 7: Previous work. Fast $\mathcal{O}(n^2)$ inversion algorithms.

Polynomial System R	Fast $\mathcal{O}(n^2)$ inversion algorithm
monomials	Traub [51]
Chebyshev polynomials	Gohberg-Olshevsky [26]
real orthogonal polynomials	Calvetti-Reichel [15]
Szegő polynomials	Olshevsky [39, 40]

The algorithms of [26, 15, 39] are not generalizations of the original Traub [51] 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], 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 system of $(H, 1)$ -quasiseparable polynomials $R = \{r_k(x)\}$ given by the generators $\{p_j, q_i, d_l, g_i, b_k, h_j\}$ of the corresponding $(H, 1)$ -quasiseparable matrix A in (2.11). Then*

$$V_R(x)^{-1} = \tilde{I} \cdot V_{\hat{R}}^T(x) \cdot \text{diag}(c_1, \dots, c_n), \quad (5.1)$$

where

$$\tilde{I} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \ddots & 1 & 0 \\ 0 & \ddots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}, \quad c_i = \prod_{\substack{k=1 \\ k \neq i}}^n (x_k - x_i)^{-1}, \quad (5.2)$$

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

$$\underbrace{\begin{bmatrix} \hat{F}_k(x) \\ \hat{r}_k(x) \end{bmatrix} = \frac{1}{q_{n-k}p_{n-k+1}} \begin{bmatrix} p_{n-k+1}q_{n-k+1}b_{n-k+1} & -p_{n-k+1}h_{n-k+1} \\ q_{n-k+1}g_{n-k+1} & x - d_{n-k+1} \end{bmatrix} \begin{bmatrix} \hat{F}_{k-1}(x) \\ \hat{r}_{k-1}(x) \end{bmatrix}}_{\text{typical terms}} + \underbrace{\begin{bmatrix} 0 \\ \frac{P_{n-k}}{q_{n-k}p_{n-k+1}} \end{bmatrix}}_{\text{perturbation term}} \quad (5.3)$$

where the coefficients P_k are defined by

$$\prod_{k=1}^n (x - x_k) = P_0 r_0(x) + P_1 r_1(x) + \cdots + P_{n-1} r_{n-1}(x) + P_n r_n(x). \quad (5.4)$$

Remark 5.2. The overall cost for computing $V_R(x)^{-1}$ via (5.1) is $\mathcal{O}(n^2)$ operations. Indeed, \tilde{I} is just a permutation, and the diagonal matrix $\text{diag}(c_1, \dots, c_n)$ 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] contains a $\mathcal{O}(n^2)$ algorithm to compute P_k 's, the recurrence relations (5.3) allow us to form the matrix $V_{\hat{R}}$ in $\mathcal{O}(n^2)$ operations which results in the overall cost of $\mathcal{O}(n^2)$ 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 with 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

$\{\hat{r}_k(x)\}$ (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 of generators. **(ii)** As opposed to (2.12) the relations (5.3) contain a certain “perturbation term.”

In fact, along with (5.3) in [7] one can find 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 other two versions in [7] are based on perturbed counterparts of Szegő-type and of general three-term recurrent relations. As opposed to the algorithm based on (5.3), the other two 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].

6 Eigenvalue problems: QR algorithms

6.1 Motivation

One standard method for computing the eigendecomposition of a symmetric matrix is two-step: **(i)** reduction of the matrix in question to a similar tridiagonal form by means of Householder transformations (unitary reflections); **(ii)** running the QR algorithm on the obtained tridiagonal matrix which can be briefly described as follows. For a given (or reduced in step (i)) matrix $A^{(1)}$ one computes the QR factorization $A^{(1)} - \sigma_1 I = QR$ (where Q is orthogonal and R is upper triangular). The next iterant $A^{(2)} = RQ + \sigma_1 I$ is obviously similar to $A^{(1)}$. It is well-known (see, e.g., [50] and the references therein) that with the right choice of shifts σ_k the sequence $\{A^{(k)}\}$ rapidly converges to a diagonal matrix thus providing the desired eigenvalues. The point is that the sparse 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 [11, 9, 53, 54]; 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 [20] (non-Hessenberg quasiseparable matrices will be defined below). As opposed to its predecessors, the algorithm of [20] 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 [20] 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, rather than by a more flexible *out-of-band quasiseparable* representation. The latter 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] to derive a generalization of the algorithm of [20] 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 α , and $P = P^{-1}$. In particular, for any vector x , one can always find a matrix P such that

$$Px = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

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

$$A = \left[\begin{array}{c|cccc} a_{11} & a_{21} & a_{31} & a_{41} \\ \hline a_{21} & & & \\ a_{31} & & A_1 & \\ a_{41} & & & \end{array} \right].$$

Let P_1 be such that

$$P_1 \begin{bmatrix} a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} = \begin{bmatrix} \star \\ 0 \\ 0 \end{bmatrix}, \quad (6.1)$$

and then since A is symmetric, we have

$$\left[\begin{array}{c|cc} 1 & 0 \\ \hline 0 & P_1 \end{array} \right] \left[\begin{array}{c|cccc} a_{11} & a_{21} & a_{31} & a_{41} \\ \hline a_{21} & & & \\ a_{31} & & A_1 & \\ a_{41} & & & \end{array} \right] \left[\begin{array}{c|cc} 1 & 0 \\ \hline 0 & P_1 \end{array} \right] = \left[\begin{array}{c|cccc} a_{11} & \star & 0 & 0 \\ \hline \star & & & \\ 0 & & P_1 A_1 P_1 & \\ 0 & & & \end{array} \right],$$

and repeating this procedure on $P_1 A_1 P_1$, 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 $\max(\text{rank} A_{21}) = n_L$ and $\max(\text{rank} A_{12}) = n_U$, where the maximum is taken over all symmetric partitions of the form*

$$A = \left[\begin{array}{c|c} * & A_{12} \\ \hline A_{21} & * \end{array} \right].$$

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 4×4 example

$$A = \left[\begin{array}{cc|cc} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{21} & a_{22} & a_{32} & a_{42} \\ \hline a_{31} & a_{32} & & \\ a_{41} & a_{42} & & A_1 \end{array} \right].$$

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

$$P_2 \left(\begin{bmatrix} a_{31} \\ a_{41} \end{bmatrix} - \begin{bmatrix} a_{32} \\ a_{42} \end{bmatrix} \right) = \begin{bmatrix} \star \\ 0 \end{bmatrix},$$

and hence

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

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

$$\begin{aligned} \left[\begin{array}{c|c} I_2 & 0 \\ \hline 0 & P_2 \end{array} \right] \left[\begin{array}{cc|cc} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{21} & a_{22} & a_{32} & a_{42} \\ \hline a_{31} & a_{32} & & \\ a_{41} & a_{42} & & A_1 \end{array} \right] \left[\begin{array}{c|c} I_2 & 0 \\ \hline 0 & P_2 \end{array} \right] = \\ \underbrace{\left[\begin{array}{cc|cc} a_{11} & a_{21} & \star & \tilde{a}^T \\ a_{21} & a_{22} & \star & \tilde{a}^T \\ \hline \star & \star & & \\ \tilde{a} & \tilde{a} & & P_2 A_1 P_2 \end{array} \right]}_{(2,2)\text{-quasiseparable matrix}} = \underbrace{\left[\begin{array}{cc|cc} a_{11} & a_{21} & \star & \tilde{a}^T \\ a_{21} & a_{22} & \star & \tilde{a}^T \\ \hline \star & \star & & \\ \tilde{a} & \tilde{a} & & \end{array} \right]}_{(1,1)\text{-out-of-band quasiseparable matrix}}. \end{aligned} \quad (6.2)$$

The above reduction step creates a dependence relation in the first and second rows, above the diagonal. 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, we see that 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*

$$A = \left[\begin{array}{c|c|c} * & * & A_{13} \\ \hline * & A_{22} & * \\ \hline A_{31} & * & * \end{array} \right],$$

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 super-diagonal/below the first subdiagonal in symmetric partitions are low rank, as illustrated next.

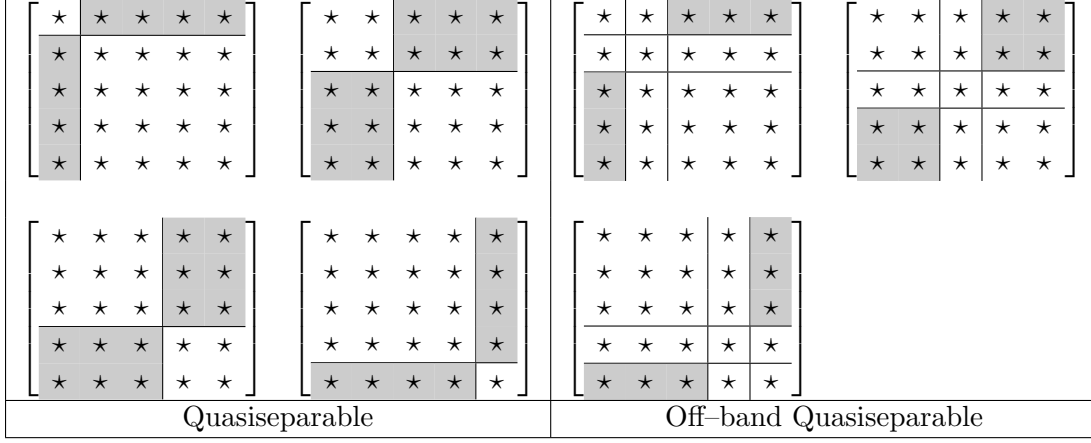


Illustration depicting which blocks have low rank in each structure

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

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

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

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

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] 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:

$$\overbrace{\underbrace{A^{(1)}}_{(r,r)}}^{\text{bandwidth}=(2m-1)} - \sigma I = \underbrace{V}_{(r,*)} \cdot \underbrace{U}_{(*,2r)} \cdot \overbrace{\underbrace{R}_{(0,2r)}}^{\text{bandwidth}=(4m-2)}$$

Here σ 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 :

$$\overbrace{\underbrace{A^{(1)}}_{(r,r)}} - \sigma I = \underbrace{Q}_{(r,2r)} \cdot \underbrace{R}_{(0,2r)}$$

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

$$\overbrace{\underbrace{A^{(2)}}_{(r,r)}}^{\text{bandwidth}=(2m-1)} = \underbrace{R}_{(0,2r)} \cdot \underbrace{Q}_{(r,2r)} + \sigma I$$

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 [14]. 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 problem from which the smaller problems were “divided.” An overview of divide-and-conquer methods was given in the introduction in Section 1.

Table 8: Divide and conquer eigenvalue algorithms.

Matrix	Algorithm
Tridiagonal	Cuppen [14], Bini–Pan [12], Gu–Eisenstat [25]
Unitary Hessenberg	Gragg–Reichel [28]
Semiseparable	Mastronardi–Van Camp–Van Barel [38]

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 below.

Remark 7.1. 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 such a (n_L, n_U) -quasiseparable matrix defined in definition 6.1. Such will be used immediately afterwards to give some notations that will be used in the divide step.

Theorem 7.2. *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 $\{p_j, q_i, d_l, g_i, b_k, h_j\}$ for $i = 1, \dots, n-1$, $j = 2, \dots, n$, $k = 2, \dots, n-1$, and $l = 1, \dots, n$, such that*

$$A = \begin{array}{|c|} \hline \begin{array}{c} d_1 \\ \vdots \\ \vdots \\ \vdots \\ p_i a_{ij}^\times q_j \\ \vdots \\ d_n \end{array} \\ \hline \end{array} \begin{array}{c} \\ \\ \\ \\ g_i b_{ij}^\times h_j \\ \\ \\ \end{array}$$

where $b_{ij}^\times = b_{i+1} \cdots b_{j-1}$ for $i > j+1$ and $b_{ij}^\times = 1$ for $i = j+1$. The generators of the matrix A are matrices of sizes

	p_k	a_k	q_k	d_k	g_k	b_k	h_k
sizes	$1 \times r'_{k-1}$	$r'_k \times r'_{k-1}$	$r'_k \times 1$	1×1	$1 \times r''_k$	$r''_{k-1} \times r''_k$	$r''_{k-1} \times 1$
range	$k \in [2, n]$	$k \in [2, n-1]$	$k \in [1, n-1]$	$k \in [1, n]$	$k \in [1, n-1]$	$k \in [2, n-1]$	$k \in [2, n]$

and these sizes are subject to

$$\max_k r'_k = n_L \quad \max_k r''_k = n_U$$

and the notation

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

Furthermore, using the notations

$$P_{m+1} = \text{col} \left(p_k a_{k,m}^\times \right)_{k=m+1}^n = \begin{bmatrix} p_{m+1} \\ p_{m+2} a_{m+1} \\ p_{m+3} a_{m+2} a_{m+1} \\ \vdots \\ p_n a_{n-1} \cdots a_{m+2} a_{m+1} \end{bmatrix} \quad (7.1)$$

$$Q_m = \text{row} \left(a_{m+1,k}^\times q_k \right)_{k=1}^m = \left[a_m \cdots a_3 a_2 q_1 \mid a_m \cdots a_3 q_2 \mid \cdots \mid a_m q_{m-1} \mid q_m \right] \quad (7.2)$$

one can partition a Hermitian quasiseparable matrix A as

$$A = \begin{bmatrix} A_m & Q_m^* P_{m+1}^* \\ P_{m+1} Q_m & B_{m+1} \end{bmatrix}. \quad (7.3)$$

7.4 New divide-and-conquer algorithm for arbitrary order quasiseparable matrices

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

$$A = \begin{bmatrix} A_m & Q_m^* P_{m+1}^* \\ P_{m+1} Q_m & B_{m+1} \end{bmatrix} = \begin{bmatrix} A'_m & 0 \\ 0 & B'_{m+1} \end{bmatrix} + \begin{bmatrix} Q_m^* \\ P_{m+1} \end{bmatrix} \begin{bmatrix} Q_m & P_{m+1}^* \end{bmatrix} \quad (7.4)$$

with⁶

$$A'_m = A_m - Q_m^* Q_m, \quad B'_{m+1} = B_{m+1} - P_{m+1} P_{m+1}^*. \quad (7.5)$$

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 this next theorem gives a more general result for non-Hermitian matrices.

Theorem 7.3. *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 $r'_m = r''_m = k$ (see Theorem 7.2), 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

$$A'_m(i, j) = A_m(i, j) - Q_m^*(i) Q_m(j) = p_i a_{i,j}^\times q_j - q_i^* (a^*)_{i,m+1}^\times a_{m+1,j}^\times q_j$$

⁶When making this calculation we are assuming that these multiplications are well-defined, which can be accomplished by padding the generators to appropriate size.

and from the equality

$$a_{m+1,j}^\times = a_m a_{m-1} \cdots a_i a_{i-1} \cdots a_{j+1} = a_{m+1,i-1}^\times a_{i,j}^\times$$

this becomes

$$A'_m(i, j) = p_i a_{ij}^\times q_j - q_i^* (a^*)_{i,m+1}^\times a_{m+1,i-1}^\times a_{i,j}^\times q_j = \left(p_i - q_i^* (a^*)_{i,m+1}^\times a_{m+1,i-1}^\times \right) a_{i,j}^\times q_j \quad (7.6)$$

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

$$A'_m(i, i) = d_i - g_i b_{i,m+1}^\times a_{m+1,j}^\times q_j$$

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. \square

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 a small-rank update, and can be solved using the common conquer step of [14].

8 Summary

In this paper we used a new concept of quasiseparable matrices to generalize the algorithms of Traub, of Björck–Pereyra and of Cuppen. These generalizations were based on the interplay between polynomial systems 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. This approach using the quasiseparable structure is unifying; special cases of the choices of generators in our algorithm correspond to special cases of quasiseparable matrices, and so this method actually generalizes all previous work.

Acknowledgements

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

References

- [1] T. Bella, *Topics in Numerical Linear Algebra related to Quasiseparable and other Structured Matrices*, PhD Thesis, University of Connecticut, advisor: Vadim Olshevsky, (available online: <http://www.tombella.com>), 2008.
- [2] T. Bella, Y. Eidelman, I. Gohberg, I. Koltracht and V. Olshevsky, *A Björck–Pereyra–type algorithm for Szegő–Vandermonde matrices based on properties of unitary Hessenberg matrices*, Linear Algebra and Its Applications, **420** Issues 2-3 (2007), 634–647.

- [3] T. Bella, Y. Eidelman, I. Gohberg, I. Koltracht and V. Olshevsky *A fast Björck–Pereyra like algorithm for solving Hessenberg-quasiseparable-Vandermonde systems*, submitted to SIAM Journal of Matrix Analysis (SIMAX), 2007.
- [4] T. Bella, Y. Eidelman, I. Gohberg, V. Olshevsky, *Classifications of three-term and two-term recurrence relations and digital filter structures via subclasses of quasiseparable matrices*, submitted to SIAM Journal of Matrix Analysis (SIMAX), 2007.
- [5] T. Bella, Y. Eidelman, I. Gohberg, V. Olshevsky, E. Tyrtyshnikov, P. Zhlobich, *A Traub-like algorithm for Hessenberg-quasiseparable-Vandermonde matrices of arbitrary order*, to appear in "Georg Heinig Memorial Volume," V.Olshevsky (Editor), to be published by Birkhauser Verlag.
- [6] T. Bella, Y. Eidelman, I. Gohberg, V. Olshevsky, *The QR iteration method for Hermitian out-of-band- quasiseparable matrices of an arbitrary order*, preprint.
- [7] T. Bella, Y. Eidelman, I. Gohberg, V. Olshevsky, E. Tyrtyshnikov *Fast Traub–like inversion algorithm for Hessenberg order one quasiseparable Vandermonde matrices*, submitted to Journal of Complexity, 2007.
- [8] T. Bella, V. Olshevsky and P. Zhlobich, *Classifications of recurrence relations via subclasses of (H,m) -quasiseparable matrices*, submitted to "Numerical Linear Algebra in Signals, Systems and Control," S. Bhattacharyya, R. Chan, P. Van Dooren, V. Olshevsky, A. Routray (Eds), to be published by Springer-Verlag.
- [9] D. A. Bini, L. Gemignani and V. Pan, *Fast and stable QR eigenvalue algorithms for generalized companion matrices and secular equations*, Numerische Mathematik, **100** (2005), 373–408.
- [10] A. Björck and V. Pereyra, *Solution of Vandermonde Systems of Equations*, Mathematics of Computation, **24** (1970), 893-903.
- [11] R. Bevilacqua and G. M. Del Corso, *Structural Properties of Matrix Unitary Reduction to Semiseparable Form*, Calcolo, **41:4** (2004), 177-202.
- [12] D.A. Bini and V. Pan, *Practical improvement of the divide-and-conquer eigenvalue algorithms*, Computing **48** (1992), 109–123.
- [13] D.A. Bini and V. Pan, *Matrix and Polynomial Computations, Vol. I: Fundamental Algorithms*, Birkhauser, Boston, 1994.
- [14] J. Cuppen, *A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem*, Numerische Mathematik, **36** (1981), 177–195.
- [15] D. Calvetti and L. Reichel, *Fast inversion of Vandermonde-like matrices involving orthogonal polynomials*, BIT, **33** (1993), 473–485.
- [16] S. Chandrasekaran , M. Gu , J. Xia and J. Zhu, *A Fast QR Algorithm for Companion Matrices*, in "Recent Advances in Matrix and Operator Theory," V. Olshevsky (Editor), Operator Theory: Advances and Applications, **179**, Birkhauser Basel, 2007, pp. 111–143.
- [17] K. B. Datta and B. M. Mohan, *Orthogonal Functions in Systems and Control*, World Scientific, 1995.

- [18] P. Dewilde and A. J. Van der Veen, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, 1998.
- [19] Y. Eidelman and I. Gohberg, *On a new class of structured matrices*, Integral Equations and Operator Theory **34** (1999), 293–324.
- [20] Y. Eidelman, I. Gohberg and V. Olshevsky, *The QR iteration method for Hermitian quasi-separable matrices of an arbitrary order*, Linear Algebra and Its Applications, **404** (2005), 305–324.
- [21] Y. Eidelman, I. Gohberg and V. Olshevsky, *Eigenstructure of Order-One-Quasiseparable Matrices. Three-term and Two-term Recurrence Relations*, Linear Algebra and its Applications, **405** (2005), 1–40.
- [22] D. Fasino, *Rational Krylov matrices and QR steps on Hermitian diagonal-plus-semiseparable matrices*, Numerical Linear Algebra and its Applications, **12** (2005), 743–754.
- [23] D. Fasino, L. Gemignani, *Direct and inverse eigenvalue problems for diagonal plus semiseparable matrices*, Numerical Algorithms, **34** (2003), 313–324.
- [24] W. B. Gragg, *Positive definite Toeplitz matrices, the Arnoldi process for isometric operators, and Gaussian quadrature on the unit circle* (in Russian). In : E.S. Nikolaev (Ed.), *Numerical methods in Linear Algebra*, pp. 16–32, Moscow University Press, 1982.
English translation in : J. Comput. and Appl. Math., **46**(1993), 183–198.
- [25] M. Gu and S. Eisenstat, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., **16** (1995), 172–191.
- [26] I. Gohberg and V. Olshevsky, *Fast inversion of Chebyshev-Vandermonde matrices*, Numerische Mathematik, **67**, No. 1 (1994), 71–92.
- [27] I. Gohberg and V. Olshevsky, *The fast generalized Parker-Traub algorithm for inversion of Vandermonde and related matrices*, J. of Complexity, **13**(2) (1997), 208–234.
- [28] W. Gragg and L. Reichel, *A Divide and Conquer Method for Unitary and Orthogonal Eigenproblems*, Numerische Mathematik, **57** (1990), 695–718.
- [29] U. Grenader and G. Szegő, *Toeplitz forms and Applications*, University of California Press, 1958.
- [30] N. J. Higham, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl., **11:1** (1990), 23–41.
- [31] G. Heinig and K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Birkhauser, 1984.
- [32] T. Kailath and V. Olshevsky *Displacement structure approach to discrete trigonometric transform based preconditioners of G. Strang and T. Chan types*, SIAM Journal on Matrix Analysis and Applications, **26**, Number 3 (2005), 706–734.
- [33] *Fast Reliable Algorithms for Matrices with Structure*, T. Kailath, A. Sayed (editors), SIAM, PA, 1999.
- [34] D. Jackson, *Fourier Series and Orthogonal Polynomials*, Dover, New York, 2004.

- [35] T. Kailath, *A theorem of I. Schur and its Impact on modern signal processing*, in Operator Theory: Advances and Applications, Gohberg (Editor), **18**, Birkhauser, Boston, 1986, pp. 9–30.
- [36] T. Kailath, *Signal processing applications of some moment problems*, in Moments in Mathematics, American Mathematical Society, Landau, editor, vol. 37, 1987, pp. 71–109.
- [37] T. Kailath, S.-Y. Kung, and M. Morf, *Displacement Ranks of Matrices and Linear Equations*, J. Math. Anal. Appl., **68/2** (1979), 395–407.
- [38] N. Mastronardi, E. Van Camp, and M. Van Barel *Divide and conquer algorithms for computing the eigendecomposition of symmetric diagonal-plus-semiseparable matrices*, Numerical Algorithms, **9**, (2005), 379–398.
- [39] V. Olshevsky, *Eigenvector computation for almost unitary Hessenberg matrices and inversion of Szego-Vandermonde matrices via Discrete Transmission lines*. Linear Algebra and Its Applications, 285 (1998), 37–67.
- [40] V. Olshevsky, *Unitary Hessenberg matrices and the generalized Parker-Forney-Traub algorithm for inversion of Szegő-Vandermonde matrices*, in Structured Matrices: Recent Developments in Theory and Computation, (D.Bini, E. Tyrtyshnikov, P. Yalamov., Eds.), NOVA Science Publ., 2001, pp. 67–78.
- [41] *Structured Matrices in Mathematics, Computer Science, and Engineering*, V. Olshevsky (editor), Volumes 1 and 2, **CONM280**, AMS publications, 2001.
- [42] V. Olshevsky, *Pivoting for structured matrices and rational tangential interpolation*, in Fast Algorithms for Structured Matrices: Theory and Applications, V.Olshevsky (editor), **CONM/323**, AMS publications, 2003, p. 1–75.
- [43] V.Olshevsky and V.Pan, *A unified superfast algorithm for boundary rational tangential interpolation problems and for inversion and factorization of dense structured matrices*, Proc. of 39th Annual Symposium on Foundations of Computer Science (FOCS’98), IEEE Computer Society, Los Alamitos, CA, 1998, pp. 192–201.
- [44] V. Olshevsky and A. Shokrollahi, *A displacement structure approach to efficient decoding of Reed-Solomon and algebraic geometric codes*, A conference paper appeared in Proc of the Thirty First ACM Symposium on Theory of Computing (STOC’99), ACM, New York, 1999, pp. 235–244.
- [45] V. Olshevsky and A. Shokrollahi, *Fast matrix-vector multiplication algorithms for confluent Cauchy-like matrices with applications*, in Proc of the Thirty Second ACM Symposium on Theory of Computing (STOC’00) ACM, New York, 2000, pp. 573–581.
- [46] B. Porat, *A Course in Digital Signal Processing*, Wiley, 1996.
- [47] V. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms*, Birkhauser, 2001.
- [48] L. Reichel and G. Opfer, *Chebyshev-Vandermonde systems*, Math. of Comp., **57** (1991), 703–721.
- [49] G. Szegő, *Orthogonal Polynomials*, AMS Publications, 1991.

- [50] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* , Springer; 3rd ed., 2002.
- [51] J. Traub, *Associated polynomials and uniform methods for the solution of linear problems*, SIAM Review, **8**, No. **3** (1966), 277–301.
- [52] E. E. Tyrtyshnikov. *How bad are Hankel matrices?* Numerische Mathematik, **67** (1994), 261–269.
- [53] M. Van Barel, R. Vandebril and N. Mastronardi, *An orthogonal similarity reduction of a matrix into semiseparable form*, SIAM J. Matrix Anal. Appl., **27:1** (2005), 176-197.
- [54] R. Vandebril, M. Van Barel and N. Mastronardi, *An implicit QR algorithm for symmetric semiseparable matrices*, Numerical Linear Algebra Appl., **12:7** (2005), 659-682.
- [55] R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices: Linear Systems*, The Johns Hopkins University Press; December 18, 2007.