A Traub–like algorithm for Hessenberg-quasiseparable-Vandermonde matrices of arbitrary order

T.Bella*, Y.Eidelman†, I.Gohberg†, V.Olshevsky*, E.Tyrtyshnikov‡ and P.Zhlobich*

Abstract. Although Gaussian elimination uses $\mathcal{O}(n^3)$ operations to invert an arbitrary matrix, matrices with a special Vandermonde structure can be inverted in only $\mathcal{O}(n^2)$ operations by the *fast* Traub algorithm. The original version of Traub algorithm was numerically unstable although only a minor modification of it yields a high accuracy in practice. The Traub algorithm has been extended from Vandermonde matrices involving monomials to polynomial–Vandermonde matrices involving real orthogonal polynomials, and the Szegö polynomials.

In this paper we consider a new more general class of polynomials that we suggest to call Hessenberg order m quasiseparable polynomials, or (H,m)–quasiseparable polynomials. The new class is wide enough to include all of the above important special cases, e.g., monomials, real orthogonal polynomials and the Szegö polynomials, as well as new subclasses. We derive a fast $\mathcal{O}(n^2)$ Traub–like algorithm to invert the associated (H,m)*–quasiseparable–Vandermonde matrices.*

The class of *quasiseparable matrices* is garnering a lot of attention recently; it has been found to be useful in designing a number of fast algorithms. The derivation of our new Traub–like algorithm is also based on exploiting quasiseparable structure of the corresponding Hessenberg matrices. Preliminary numerical experiments are presented comparing the algorithm to standard structure ignoring methods.

This paper extends our recent results in [6] from the $(H,0)$– and $(H,1)$–quasiseparable cases to the more general (H,m)–quasiseparable case.

1. Introduction. Polynomial–Vandermonde matrices and quasiseparable matrices.

1.1. Inversion of polynomial–Vandermonde matrices.

In this paper we consider the problem of inverting the class of polynomial–Vandermonde matrices. For a set of n distinct nodes $\{x_k\}_{k=1}^{n}$, the classical Vandermonde matrix $V(x) = \left[\ x_i^{j-1}\ \right]$ is known to be invertible (provided the nodes are distinct). One can generalize this structure by evaluating a different basis (other than the monomials) at the nodes in the following way. That is, for a set of n polynomials $R = \{r_0(x), r_1(x), \ldots, r_{n-1}(x)\}$ satisfying $\deg r_k(x) = k$, the matrix of the form

$$V_R(x) = \begin{bmatrix} r_0(x_1) & r_1(x_1) & \cdots & r_{n-1}(x_1) \\ r_0(x_2) & r_1(x_2) & \cdots & r_{n-1}(x_2) \\ \vdots & \vdots & & \vdots \\ r_0(x_n) & r_1(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix} \tag{1.1}$$

is called a polynomial–Vandermonde matrix. It is clear that such a matrix is invertible if and only if the chosen nodes are distinct. Indeed, let T be an invertible, upper triangular matrix, and consider the product $V(x)\cdot T$. The effect of post–multiplication by T is that the entries of the product are polynomials determined

0* Department of Mathematics, University of Connecticut, Storrs CT 06269-3009, USA. Email: bella@math.uconn.edu (contact author), olshevsky@math.uconn.edu
0† School of Mathematical Sciences, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Ramat-Aviv 69978, Israel. Email: eideyu@post.tau.ac.il, gohberg@post.tau.ac.il
0‡ Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street, 8, Moscow, 119991, Russia. Email: tee@inm.ras.ru

by the columns of T, evaluated at the given nodes. Hence this is an alternate definition of a polynomial–Vandermonde matrix as the product of invertible matrices (provided the nodes are distinct).

In the simplest case where $R = \{1, x, x^2, \ldots, x^{n-1}\}$ (i.e. when $T = I$), the matrix $V_R(x)$ reduces to a classical Vandermonde matrix and the inversion algorithm is due to Traub [28]. It was observed in [16] that a minor modification of the original Traub algorithm results in very good accuracy.

The structure–ignoring approach of Gaussian elimination for inversion of $V_R(x)$ requires $\mathcal{O}(n^3)$ operations, and for a general matrix $V_R(x)$ (i.e. no special recurrence relations satisfied by the polynomial system R involved), the algorithm derived in this paper also requires $\mathcal{O}(n^3)$ operations. However, in several special cases, the structure has been exploited, resulting in fast algorithms that can compute the n^2 entries of the inverse in only $\mathcal{O}(n^2)$ operations. It also allows the construction of fast system solvers; one of the pioneering works in this area belongs to Björck and Pereyra [2]. Table 1 lists the previous work in deriving fast inversion algorithms and fast system solvers for various special cases of the polynomial system R.

Table 1. Fast $\mathcal{O}(n^2)$ algorithms for polynomial–Vandermonde matrices.

Matrix $V_R(x)$	Polynomial System R	$\mathcal{O}(n^2)$ inversion	$\mathcal{O}(n^2)$ system solver
Classical Vandermonde	monomials	Traub [28]	Björck-Pereyra [2]
Chebyshev-Vandermonde	Chebyshev polynomials	Gohberg-Olshevsky [14]	Reichel-Opfer [26]
Three-Term Vandermonde	Real orthogonal polynomials	Calvetti-Reichel [7]	Higham [19]
Szegö-Vandermonde	Szegö polynomials	Olshevsky [23]	BEGKO [3]

1.2. Capturing recurrence relations via confederate matrices.

To generalize the inversion algorithms of Table 3 we will use the concept of a *confederate matrix* introduced in [21]. Let polynomials $R = \{r_0(x), r_1(x), \ldots, r_n(x)\}$ be specified by the general n-term recurrence relations[1]

$$r_k(x) = (\alpha_k x - a_{k-1,k}) \cdot r_{k-1}(x) - a_{k-2,k} \cdot r_{k-2}(x) - \ldots - a_{0,k} \cdot r_0(x), \quad \alpha_k \neq 0 \tag{1.2}$$

for $k > 0$, and r_0 is a constant. Define for the polynomial

$$P(x) = P_0 \cdot r_0(x) + P_1 \cdot r_1(x) + \ldots + P_{n-1} \cdot r_{n-1}(x) + P_n \cdot r_n(x) \tag{1.3}$$

its *confederate matrix* (with respect to the polynomial system R) by

$$C_R(P) = \underbrace{\begin{bmatrix} \frac{a_{01}}{\alpha_1} & \frac{a_{02}}{\alpha_2} & \frac{a_{03}}{\alpha_3} & \cdots & \frac{a_{0,k}}{\alpha_k} & \cdots & \cdots & \frac{a_{0,n}}{\alpha_n} \\ \frac{1}{\alpha_1} & \frac{a_{12}}{\alpha_2} & \frac{a_{13}}{\alpha_3} & \cdots & \frac{a_{1,k}}{\alpha_k} & \cdots & \cdots & \frac{a_{1,n}}{\alpha_n} \\ 0 & \frac{1}{\alpha_2} & \frac{a_{23}}{\alpha_3} & \cdots & \vdots & \cdots & \cdots & \frac{a_{2,n}}{\alpha_n} \\ 0 & 0 & \frac{1}{\alpha_3} & \ddots & \frac{a_{k-2,k}}{\alpha_k} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{a_{k-1,k}}{\alpha_k} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{1}{\alpha_k} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{a_{n-1,n}}{\alpha_n} \end{bmatrix}}_{C_R(r_n)} - \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_{n-1} \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 & \frac{1}{\alpha_n P_n} \end{bmatrix} \tag{1.4}$$

In the special case where $P(x) = r_n(x)$, we have $P_0 = P_1 = \cdots = P_{n-1} = 0$, and hence the last term on the right hand side of (1.4) vanishes.

Notice that the coefficients of the recurrence relations for the k^{th} polynomial $r_k(x)$ from (1.2) are contained in the k^{th} column of $C_R(r_n)$, as the highlighted column shows. We refer to [21] for many useful properties of the confederate matrix and only recall here that

$$r_k(x) = \alpha_0 \cdot \alpha_1 \cdot \ldots \cdot \alpha_k \cdot \det(xI - [C_R(P)]_{k \times k}),$$

[1] It is easy to see that any polynomial system $\{r_k(x)\}$ satisfying $\deg r_k(x) = k$ obeys (1.2).

and

$$P(x) = \alpha_0 \cdot \alpha_1 \cdots \alpha_n \cdot P_n \cdot \det(xI - C_R(P)),$$

where $[C_R(P)]_{k\times k}$ denotes the $k \times k$ leading submatrix of $C_R(P)$ in the special case where $P(x) = r_n(x)$.

Next in Table 2 we list confederate matrices for the polynomial systems[2] of Table 3.

TABLE 2. Systems of polynomials and corresponding recurrence relations.

Polynomial System Recurrence relations	Corresponding confederate matrix $C_R(r_n)$
monomials $r_k(x) = x \cdot r_{k-1}(x)$	$\begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & \ddots & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$ lower shift matrix
Chebyshev polynomials $r_k(x) = 2x \cdot r_{k-1}(x) - r_{k-2}(x)$	$\begin{bmatrix} 0 & \frac{1}{2} & \cdots & \cdots & 0 \\ \frac{1}{2} & \ddots & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{1}{2} \\ 0 & \cdots & 0 & \frac{1}{2} & 0 \end{bmatrix}$ tridiagonal matrix
real orthogonal polynomials $r_k(x) = (\alpha_k x - \delta_k) r_{k-1}(x) - \gamma_k \cdot r_{k-2}(x)$	$\begin{bmatrix} \frac{\delta_1}{\alpha_1} & \frac{\gamma_2}{\alpha_2} & 0 & \cdots & 0 \\ \frac{1}{\alpha_1} & \frac{\delta_2}{\alpha_2} & \ddots & \ddots & \vdots \\ 0 & \frac{1}{\alpha_2} & \ddots & \frac{\gamma_{n-1}}{\alpha_{n-1}} & 0 \\ \vdots & \ddots & \ddots & \frac{\delta_{n-1}}{\alpha_{n-1}} & \frac{\gamma_n}{\alpha_n} \\ 0 & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{\delta_n}{\alpha_n} \end{bmatrix}$ tridiagonal matrix
Szegö polynomials[3] $\begin{bmatrix} \phi_k(x) \\ \phi_k^{\#}(x) \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} 1 & -\rho_k^* \\ -\rho_k & 1 \end{bmatrix} \begin{bmatrix} \phi_{k-1}(x) \\ x\phi_{k-1}^{\#}(x) \end{bmatrix}$	$\begin{bmatrix} -\rho_1\rho_0^* & \cdots & -\rho_{n-1}\mu_{n-2}\cdots\mu_1\rho_0^* & -\rho_n\mu_{n-1}\cdots\mu_1\rho_0^* \\ \mu_1 & \ddots & -\rho_{n-1}\mu_{n-2}\cdots\mu_2\rho_1^* & -\rho_n\mu_{n-1}\cdots\mu_2\rho_1^* \\ 0 & \ddots & \vdots & \vdots \\ \vdots & & -\rho_{n-1}\rho_{n-2}^* & -\rho_n\mu_{n-1}\rho_{n-2}^* \\ 0 & \cdots & \mu_{n-1} & -\rho_n\rho_{n-1}^* \end{bmatrix}$ unitary Hessenberg matrix matrix

It turns out that all matrices of Table 2 are special cases of the more general class of matrices defined next. It is this larger class of matrices, and the class of polynomials related to them via (1.2) that we consider in this paper.

---

[2]For the monomials, Chebyshev polynomials and real orthogonal polynomials the structure of the confederate matrices can be immediately deduced from their recurrence relations. For Szegö polynomials it is also well-known, see, e.g., [23] and the references therein.

[3]It is known that, under the additional restriction of $\rho_k \neq 0$ for each k, the corresponding Szegö polynomials satisfy the three-term recurrence relations

$$\phi_0^{\#}(x) = \frac{1}{\mu_0}, \quad \phi_1^{\#}(x) = \frac{1}{\mu_1}(x \cdot \phi_0^{\#}(x) + \rho_1\rho_0^* \cdot \phi_0^{\#}(x))$$
$$\phi_k^{\#}(x) = \left[\frac{1}{\mu_k} \cdot x + \frac{\rho_k}{\rho_{k-1}} \frac{1}{\mu_k}\right] \phi_{k-1}^{\#}(x) - \frac{\rho_k}{\rho_{k-1}} \frac{\mu_{k-1}}{\mu_k} \cdot x \cdot \phi_{k-2}^{\#}(x)$$

1.3. Main tool: quasiseparable matrices and polynomials.

Definition 1.1. (Quasiseparable matrices and polynomials)

- *A matrix A is called (H,m)–quasiseparable (i.e., Hessenberg lower part and order m upper part) if* **(i)** *it is strongly upper Hessenberg (i.e. nonzero first subdiagonal, $a_{i+1,i} \neq 0$), and* **(ii)** $\max(\operatorname{rank} A_{12}) = m$, *where the maximum is taken over all symmetric partitions of the form*

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

- *Let $A = [a_{ij}]$ be a (H,m)–quasiseparable matrix. For $\alpha_i = 1/a_{i+1,i}$, then the system of polynomials related to A via*

$$r_k(x) = \alpha_1 \cdots \alpha_k \det (xI - A)_{(k \times k)} .$$

  is called a system of (H,m)–quasiseparable polynomials.

Remark 1.2. The class of (H,m)–quasiseparable polynomials is wide enough to include monomials, Chebyshev polynomials, real orthogonal and Szegö polynomials (i.e., all polynomials of Tables 3 and 2) as special cases. This can be seen by inspecting, for each confederate matrix, its typical submatrix A_{12} from the partition described in Definition 1.1.

- **The lower shift matrix is $(H,0)$–quasiseparable** Indeed, if A is such a matrix, then any submatrix A_{12} is simply a zero matrix.
- **Tridiagonal matrices are also $(H,1)$–quasiseparable**. Indeed, if A is tridiagonal, then the submatrix A_{12} has the form $(\gamma_j/\alpha_j)e_k e_1^T$, which can easily be observed to have rank one.
- **Unitary Hessenberg matrices are $(H,1)$–quasiseparable**. Indeed, if A corresponds to the Szegö polynomials, then the corresponding $3 \times (n-1)$ submatrix A_{12} has the form

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

  which is also rank 1 since the rows are scalar multiples of each other. The same is true for all other symmetric partitions of A.

Hence all of the polynomials corresponding to the confederate matrices listed above are $(H,1)$–quasiseparable polynomials.

1.4. Main problem: Inversion of (H,m)–quasiseparable–Vandermonde matrices.

As shown in the previous remark, $(H,0)$– and $(H,1)$–quasiseparable polynomials generalize the previous cases of monomials, real orthogonal polynomials, and Szegö polynomials. In the paper [6], an algorithm for inversion of $(H,0)$– and $(H,1)$–quasiseparable–Vandermonde matrices is derived, and hence that algorithm is applicable to these special cases. However there are important cases not covered by either $(H,0)$– or $(H,1)$–quasiseparable polynomials. An example of such a system of polynomials is given next.

Example 1.3 (l-recurrent polynomials). From (1.4) it follows that if polynomials satisfy l-term recurrence relations

$$r_k(x) = (\alpha_k x - a_{k-1,k}) \cdot r_{k-1}(x) - a_{k-2,k} \cdot r_{k-2}(x) - \ldots - a_{k-(l-1),k} \cdot r_{k-(l-1)}(x) \tag{1.5}$$

then their confederate matrices

$$A = \begin{bmatrix} \frac{a_{0,1}}{\alpha_1} & \cdots & \frac{a_{0,l-1}}{\alpha_{l-1}} & 0 & \cdots & 0 \\ \frac{1}{\alpha_1} & \frac{a_{1,2}}{\alpha_2} & \cdots & \frac{a_{1,l}}{\alpha_l} & \ddots & \vdots \\ 0 & \frac{1}{\alpha_2} & & & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \frac{a_{n-(l-1),n}}{\alpha_n} \\ \vdots & & \ddots & \frac{1}{\alpha_{n-2}} & & \vdots \\ 0 & \cdots & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{a_{n-1,n}}{\alpha_n} \end{bmatrix} \tag{1.6}$$

are $(1, l-2)$-banded, i.e., they have only one nonzero subdiagonal and $l-2$ nonzero superdiagonals. Considering a typical element A_{12} of the partition of Definition 1.1, in this case for a 5×5, $(1,2)$-banded example, we have

$$A = \left[\begin{array}{c|c} * & A_{12} \\ \hline * & * \end{array}\right] = \left[\begin{array}{cc|ccc} \frac{a_{0,1}}{\alpha_1} & \frac{a_{0,2}}{\alpha_2} & \frac{a_{0,3}}{\alpha_3} & 0 & 0 \\ \frac{1}{\alpha_1} & \frac{a_{1,2}}{\alpha_2} & \frac{a_{1,3}}{\alpha_3} & \frac{a_{1,4}}{\alpha_4} & 0 \\ \hline 0 & \frac{1}{\alpha_2} & \frac{a_{2,3}}{\alpha_3} & \frac{a_{2,4}}{\alpha_4} & \frac{a_{2,5}}{\alpha_5} \\ 0 & 0 & \frac{1}{\alpha_3} & \frac{a_{3,4}}{\alpha_4} & \frac{a_{3,5}}{\alpha_5} \\ 0 & 0 & 0 & \frac{1}{\alpha_4} & \frac{a_{4,5}}{\alpha_5} \end{array}\right]$$

One can see that any A_{12} of the partition of Definition 1.1 has rank at most 2, implying that A is a $(H,2)$–quasiseparable matrix by definition. More generally, a system of l-recurrent polynomials are $(H, l-2)$–quasiseparable (and so polynomials satisfying three-term recurrence relations are $(H,1)$–quasiseparable).

A Björck–Pereyra–like algorithm for solving linear systems with Hessenberg–quasiseparable–Vandermonde coefficient matrices was proposed in [4], and this algorithm is applicable for any order of quasiseparability (i.e. for any $m \geqslant 1$ of (H,m)–quasiseparable matrix). A Traub–like inversion algorithm is derived in [6], but it is valid only for $(H,0)$– and $(H,1)$–quasiseparable–Vandermonde matrices. In this paper we extend the Traub–like algorithm to a corresponding algorithm for an arbitrary order of quasiseparability. Previous work in this area, including that using quasiseparable matrices, is given next in Table 3.

TABLE 3. Fast $\mathcal{O}(n^2)$ algorithms for polynomial–Vandermonde matrices.

Matrix $V_R(x)$	Polynomial System R	$\mathcal{O}(n^2)$ inversion	$\mathcal{O}(n^2)$ system solver
Classical Vandermonde	monomials	Traub [28]	Björck-Pereyra [2]
Chebyshev-Vandermonde	Chebyshev polynomials	Gohberg-Olshevsky [14]	Reichel-Opfer [26]
Three-Term Vandermonde	Real orthogonal polynomials	Calvetti-Reichel [7]	Higham [19]
Szegö-Vandermonde	Szegö polynomials	Olshevsky [23]	BEGKO [3]
$(H,1)$–quasiseparable–V	$(H,1)$–quasiseparable	BEGOT [6]	BEGKO [4]
(H,m)–quasiseparable–V	(H,m)–quasiseparable	this paper	

This new inversion algorithm is applicable to the special cases of polynomial–Vandermonde matrices for monomials, real orthogonal polynomials, and Szegö polynomials, which are themselves special cases of (H,m)–quasiseparable polynomials. In addition, it is also applicable to new classes of polynomials for which no Traub–like algorithm is currently available. One such class of polynomials are those satisfying the motivating recurrence relations (1.5).

As was the case for the Traub–like algorithm for $(H,0)$– and $(H,1)$–quasiseparable-Vandermonde matrices of [6], the proposed Traub–like algorithm for (H,m)–quasiseparable-Vandermonde matrices is fast, requiring only $\mathcal{O}(n^2)$ operations by exploiting the sparse recurrence relations (1.5).

2. Inversion formula

In this section we recall the formula that will be used to invert a polynomial-Vandermonde matrix as in (1.1). Such a matrix is completely determined by n polynomials $R = \{r_0(x), \ldots, r_{n-1}(x)\}$ and n nodes $x = (x_1, \ldots, x_n)$. The desired inverse $V_R(x)^{-1}$ is given by the formula

$$V_R(x)^{-1} = \tilde{I} \cdot V_R^T(x) \cdot \mathrm{diag}(c_1, \ldots, c_n), \tag{2.1}$$

(see [22], [23]) where

$$c_i = \prod_{\substack{k=1 \\ k \neq i}}^{n} (x_k - x_i)^{-1}, \tag{2.2}$$

$\tilde{I}$ is the antidiagonal matrix

$$\tilde{I} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & 1 & 0 \\ 0 & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}, \tag{2.3}$$

and $\widehat{R}$ is the system of *associated (generalized Horner) polynomials*, defined as follows: if we define the *master polynomial* $P(x)$ by $P(x) = (x-x_1)\cdots(x-x_n)$, then for the polynomial system $R = \{r_0(x), \dots, r_{n-1}(x), P(x)\}$, the associated polynomials $\widehat{R} = \{\widehat{r}_0(x), \dots, \widehat{r}_{n-1}(x), P(x)\}$ are those satisfying the relations

$$\frac{P(x)-P(y)}{x-y} = \sum_{k=0}^{n-1} r_k(x) \cdot \widehat{r}_{n-k-1}(y), \tag{2.4}$$

see [20]. A discussion showing the existence of polynomials satisfying these relations (2.4) for any polynomial system R is given in [3]. This definition can be seen as a generalization of the Horner polynomials associated with the monomials, cf. with the discussion in Section 2.1 below.

This discussion gives a relation between the inverse $V_R(x)^{-1}$ and the polynomial-Vandermonde matrix $V_{\widehat{R}}(x)$, where $\widehat{R}$ is the system of polynomials associated with R. To use this in order to invert $V_R(x)$, one needs to evaluate the polynomials $\widehat{R}$ at the nodes x to form $V^T_{\widehat{R}}(x)$. Such evaluation can be done by using confederate matrices (defined in Section 1.2) associated with system of polynomials, which will be discussed in the next section, but at this point the formula (2.1) allows us to present a sketch of the Traub–like inversion algorithm next. The detailed algorithm will be provided in Section 6 below after deriving next several formulas that will be required to implement its steps 2 and 3.

Algorithm 2.1. *[A sketch of the Traub–like inversion algorithm]*

1. *Compute the entries of* $diag(c_1, \dots, c_n)$ *via* (2.2).
2. *Compute the coefficients* $\{P_0, P_1, \dots, P_n\}$ *of the master polynomial* $P(x)$ *as in* (1.3).
3. *Evaluate the* n *polynomials of* $\widehat{R}$ *with confederate matrix specified via* (2.8) *at the* n *nodes* x_k *to form* $V_{\widehat{R}}(x)$.
4. *Compute the inverse* $V_R(x)^{-1}$ *via* (2.1).

2.1. The key property of all Traub–like algorithms: pertransposition

In this section we use the classical Traub algorithm to explain the key property used in deriving a Traub–like algorithms in terms of quasiseparable ranks of confederate matrices of systems of polynomials. According to (1.3), let

$$P(x) = P_0 + P_1 \cdot x + \dots + P_{n-1} \cdot x^{n-1} + x^n$$

be a polynomial in the monomial base. The monomials $R = \{1, x, x^2, \dots, x^{n-1}\}$ satisfy the obvious recurrence relations $x^k = x \cdot x^{k-1}$ and hence the confederate matrix (1.4) of $P(x)$ with respect to R becomes

$$C_R(P) = \begin{bmatrix} 0 & 0 & \cdots & 0 & -P_0 \\ 1 & 0 & \cdots & 0 & -P_1 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -P_{n-1} \end{bmatrix} \tag{2.5}$$

which is the well–known companion matrix. By Definition 1.1 its leading submatrices $[C_R(P)]_{k\times k}$ are $(H, 0)$–quasiseparable for $k = 1 \dots n-1$. Hence monomials are $(H, 0)$–quasiseparable polynomials.

From the well–known recurrence relations for the Horner polynomials (which invert the classical Vandermonde matrix, see [28])

$$\widehat{r}_0(x) = 1, \quad \widehat{r}_k(x) = x \cdot \widehat{r}_{k-1}(x) + P_{n-k}, \tag{2.6}$$

we obtain the confederate matrix

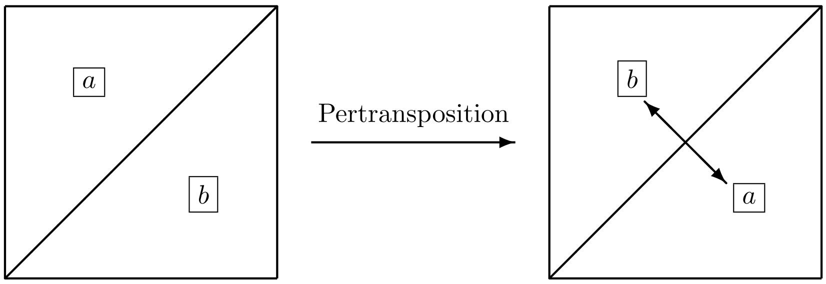
$$C_{\widehat{R}}(P) = \begin{bmatrix} -P_{n-1} & -P_{n-2} & \cdots & -P_1 & -P_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}. \tag{2.7}$$

for the Horner polynomials. This relation between the confederate matrices of R and $\widehat{R}$ can be seen as

$$C_{\widehat{R}}(P) = \tilde{I} \cdot C_R(P)^T \cdot \tilde{I}, \tag{2.8}$$

or more visually,

$C_R(P) =$



$= C_{\widehat{R}}(P),$

and holds in the general case (see [22], [23]). The passage from $C_R(P)$ to $C_{\widehat{R}}(P)$ in (2.8) is called a *pertransposition*, or reflection across the antidiagonal. We will show later that recurrence relations for a given system of polynomials together with (2.8) allow fast evaluation of the polynomials $\widehat{R}$ at the nodes x, a required step for a fast Traub–like algorithm.

The leading submatrices of the matrix (2.7) are easily seen to be Hessenberg and quasiseparable[4] but due to the perturbation $\{P_0, P_1, \ldots, P_{n-1}\}$ their quasiseparable ranks are 1. Hence the Horner polynomials are $(H, 1)$–quasiseparable; that is, the quasiseparable rank increases by one due to the inclusion of the perturbation terms in each principal submatrix. Analogously, for an arbitrary system of (H, m)–quasiseparable polynomials the order of quasiseparability increases by at most one, which we state as follows.

Remark 2.2. For R a system of (H, m)–quasiseparable polynomials, when passing to the system $\widehat{R}$ of polynomials associated with R, the order of quasiseparability may increase by one. That is, the system $\widehat{R}$ is either (H, m)–quasiseparable or $(H, m+1)$–quasiseparable.

This property is used by all of the previous Traub–like algorithms given in Table 3. The quasiseparable rank of a confederate matrix after pertransposition is increased only by at most one. This allows derivation of cheap recurrence relations for the system of polynomials $\widehat{R}$ and use them in computing $V_{\widehat{R}}$.

In summary, the classical Traub algorithm was based on deriving formulas for systems of polynomials as the confederate matrices changed from $(H, 0)$–quasiseparable to $(H, 1)$–quasiseparable, and subsequent Traub–like algorithms were based on changes from $(H, 1)$–quasiseparable to $(H, 2)$–quasiseparable. Such derivations were already very involved.

In order to derive a Traub–like algorithm in the more general case considered in this paper, we need to **(i)** attain the formulas for the original (H, m)–quasiseparable polynomials which, unlike previous cases, are not readily available[5], and **(ii)** have a derivation allowing us to pass from (H, m)–quasiseparable to $(H, m+1)$–quasiseparable confederate matrices. Hence our plan for the next two sections is to solve the problems (i) and (ii) listed above, respectively.

3. Recurrence relations for (H, m)–quasiseparable polynomials

Real orthogonal polynomials are typically defined in terms of the recurrence relations they satisfy, and then these recurrence relations are used to give equivalent definitions in terms of Hessenberg matrices, as in Table

[4]Pertransposition changes the order in which the submatrices A_{12} of Definition 1.1 appear and transposes them, but does not change their ranks, hence the quasiseparability is preserved.

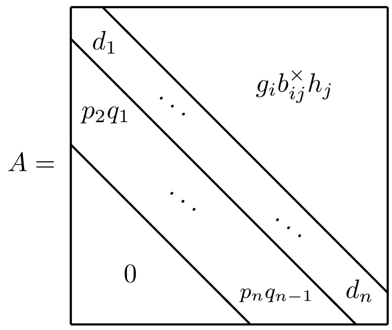
[5]In the $(H, 1)$–quasiseparable case, these formulas were derived in [6].

2. Currently we have only the latter definition for (H,m)–quasiseparable polynomials, i.e. in terms of the related (H,m)–quasiseparable matrix. Since the main tool is designing fast algorithms is the recurrence relations, as in (2.6), it is the goal of this section to derive a set of sparse recurrence relations satisfied by (H,m)–quasiseparable polynomials.

3.1. Generators of quasiseparable matrices.

We begin with an equivalent definition of quasiseparability in terms of *generators*. Such generators are the compressed representation of a quasiseparable matrix; that is, the $\mathcal{O}(n)$ entries of the generators define the $\mathcal{O}(n^2)$ entries of the matrix. Operations with generators are the key in designing various types of fast algorithms. In this case, the sparse recurrence relations for (H,m)–quasiseparable polynomials will be given in terms of these generators.

Definition 3.1 (Generator definition for (H,m)–quasiseparable matrices). *A matrix A is called (H,m)–quasiseparable if* **(i)** *it is strongly upper Hessenberg (i.e. nonzero first subdiagonal, $a_{i+1,i} \neq 0$), and* **(ii)** *it can be represented in the form*



(3.1)

where $b^{\times}_{ij} = b_{i+1} \cdots b_{j-1}$ for $j > i+1$ and $b^{\times}_{ij} = 1$ for $j = i+1$. The elements

$$\{p_k, q_k, d_k, g_k, b_k, h_k\},$$

called the generators of the matrix A, are matrices of sizes

	p_k	q_k	d_k	g_k	b_k	h_k
sizes	1×1	1×1	1×1	$1 \times u_k$	$u_{k-1} \times u_k$	$u_{k-1} \times 1$
range	$k \in [2,n]$	$k \in [1,n-1]$	$k \in [1,n]$	$k \in [1,n-1]$	$k \in [2,n-1]$	$k \in [2,n]$

subject to $\max_k u_k = m$.

Remark 3.2. For a given (H,m)–quasiseparable matrix the set of generators of Definition 3.1 is not unique. There is a freedom in choosing generators without changing the matrix.

Remark 3.3. It is useful to note that Definition 3.1 together with (1.2) imply that (H,m)–quasiseparable polynomials satisfy n-term recurrence relations

$$r_k(x) = \frac{1}{p_{k+1}q_k}\left[(x-d_k)r_{k-1}(x) - \sum_{j=0}^{k-2}\left(g_{j+1}b^{\times}_{j+1,k}h_k r_j(x)\right)\right]. \tag{3.2}$$

This formula is not sparse and hence expensive. In order to design a fast algorithm, sparse recurrence relations are required, and such are stated and proved in the next section.

3.2. Sparse recurrence relations for (H,m)–quasiseparable polynomials.

The next theorem gives, for any (H,m)–quasiseparable matrix, a set of sparse recurrence relations satisfied by the corresponding (H,m)–quasiseparable polynomials. These recurrence relations are given in terms of the generators of the (H,m)–quasiseparable matrix.

Theorem 3.4. *Let A be a (H,m)–quasiseparable matrix specified by the generators $\{p_k, q_k, d_k, g_k, b_k, h_k\}$. Then the polynomial system $R = \{r_k(x)\}_{k=0}^{n}$ corresponding to A (such that $A = C_R(r_n)$) satisfies*

$$\left[\begin{array}{c} F_k(x) \\ \hline r_k(x) \end{array}\right] = \frac{1}{p_{k+1}q_k}\left[\begin{array}{c|c} p_kq_k\, b_k^T & -q_k\, g_k^T \\ \hline p_k\, h_k^T & x-d_k \end{array}\right]\left[\begin{array}{c} F_{k-1}(x) \\ \hline r_{k-1}(x) \end{array}\right] \tag{3.3}$$

The proof is given at the end of this section, however first some special cases are given in detail.

Example 3.5 ($(H,0)$– and $(H,1)$–quasiseparable case). For the case where $m \leqslant 1$, the recurrence relations (3.3) reduce to those derived in [11], which were used in [6] to derive the Traub–like algorithm for $(H,0)$– and $(H,1)$–quasiseparable–Vandermonde matrices; that is, of the form

$$\begin{bmatrix} F_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & \delta_k x + \theta_k \end{bmatrix}\begin{bmatrix} F_{k-1}(x) \\ r_{k-1}(x) \end{bmatrix}. \tag{3.4}$$

These were referred to as [EGO05]–type recurrence relations in [6], and as the recurrence relations (3.3) are a generalization of these, we refer to (3.3) as [EGO05]–type recurrence relations as well.

A motivating example for considering the larger class of (H,m)–quasiseparable polynomials was their inclusion of the class of l–recurrent polynomials, a class not contained by previous cases.

Example 3.6 (l-recurrent polynomials). By introducing the auxiliary polynomials $f_k^{(1)}(x), \ldots f_k^{(l-2)}(x)$, the relation (1.5) can be rewritten as

$$\begin{bmatrix} f_k^{(1)}(x) \\ \vdots \\ \vdots \\ \vdots \\ f_k^{(l-2)} \\ r_k(x) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & -a_{k-2,k} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & \ddots & 0 & \vdots \\ \vdots & & & \ddots & 1 & \vdots \\ 0 & \dots & \dots & \dots & 0 & -a_{k-(l-1),k} \\ 1 & 0 & \dots & \dots & 0 & \alpha_k x - a_{k-1,k} \end{bmatrix}\begin{bmatrix} f_{k-1}^{(1)}(x) \\ \vdots \\ \vdots \\ \vdots \\ f_{k-1}^{(l-2)} \\ r_{k-1}(x) \end{bmatrix},$$

which is the reduction of (3.3) in this special case.

Proof of Theorem 3.4. The recurrence relations (3.3) define a system of polynomials which satisfy the n-term recurrence relations

$$r_k(x) = (\alpha_k x - a_{k-1,k}) \cdot r_{k-1}(x) - a_{k-2,k} \cdot r_{k-2}(x) - \ldots - a_{0,k} \cdot r_0(x) \tag{3.5}$$

for some coefficients $\alpha_k, a_{k-1,k}, \ldots, a_{0,k}$. The proof is presented by showing that these n-term recurrence relations in fact coincide exactly with (3.2), so these coefficients coincide with those of the n-term recurrence relations of the polynomials R. Using relations for $r_k(x)$ and $F_{k-1}(x)$ from (3.3), we have

$$r_k(x) = \frac{1}{p_{k+1}q_k}\left[(x-d_k)r_{k-1}(x) - g_{k-1}h_k r_{k-2}(x) + p_{k-1}h_k^T b_{k-1}^T F_{k-2}(x)\right]. \tag{3.6}$$

Notice that again using (3.3) to eliminate $F_{k-2}(x)$ from the equation (3.6) will result in an expression for $r_k(x)$ in terms of $r_{k-1}(x)$, $r_{k-2}(x)$, $r_{k-3}(x)$, $F_{k-3}(x)$, and $r_0(x)$ without modifying the coefficients of $r_{k-1}(x)$, $r_{k-2}(x)$, or $r_0(x)$. Again applying (3.3) to eliminate $F_{k-3}(x)$ results in an expression in terms of $r_{k-1}(x)$, $r_{k-2}(x)$, $r_{k-3}(x)$, $r_{k-4}(x)$, $F_{k-4}(x)$, and $r_0(x)$ without modifying the coefficients of $r_{k-1}(x)$, $r_{k-2}(x)$, $r_{k-3}(x)$, or $r_0(x)$. Continuing in this way, the n-term recurrence relations of the form (3.5) are obtained without modifying the coefficients of the previous ones.

Suppose that for some $0 < j < k-1$ the expression for $r_k(x)$ is of the form

$$r_k(x) = \frac{1}{p_{k+1}q_k}[(x-d_k)r_{k-1}(x) - g_{k-1}h_k r_{k-2}(x) - \cdots \\ - g_{j+1}b_{j+1,k}^{\times} h_k r_j(x) + p_{j+1}h_k^T (b_{j,k}^{\times})^T F_j(x)]. \tag{3.7}$$

Using (3.3) for $F_j(x)$ gives the relation

$$F_j(x) = \frac{1}{p_{j+1}q_j}\left(p_j q_j b_j^T F_{j-1}(x) - q_j g_j^T r_{j-1}(x)\right) \tag{3.8}$$

Inserting (3.8) into (3.7) gives

$$r_k(x) = \frac{1}{p_{k+1}q_k}\left[(x-d_k)r_{k-1}(x) - g_{k-1}h_k r_{k-2}(x) - \cdots - g_j b_{j,k}^{\times} h_k r_{j-1}(x) + p_j h_k^T \left(b_{j-1,k}^{\times}\right)^T F_{j-1}(x)\right].$$

Therefore since (3.6) is the case of (3.7) for $j = k-2$, (3.7) is true for each $j = k-2, k-3, \ldots, 0$, and for $j = 0$, using the fact that $F_0 = 0$ we have

$$r_k(x) = \frac{1}{p_{k+1}q_k}\left[(x-d_k)r_{k-1}(x) - g_{k-1}h_k r_{k-2}(x) - \cdots - g_1 b_{1,k}^{\times} h_k r_0(x)\right]$$

Since these coefficients coincide with (3.2) that are satisfied by the polynomial system R, the polynomials given by (3.3) must coincide with these polynomials. This proves the theorem. □

4. Recurrence relations for polynomials associated with (H,m)–quasiseparable polynomials

In the previous section, sparse recurrence relations for (H,m)–quasiseparable polynomials were derived. However, the classical Traub algorithm is based on the recurrence relations for the Horner polynomials, not the original monomial system. In this section, sparse recurrence relations for the system of polynomials associated with a system of (H,m)–quasiseparable polynomials (i.e. the generalized Horner polynomials) are derived. It is these recurrence relations that form the basis of the Traub–like algorithm.

4.1. Introduction of a perturbation term via pertransposition of confederate matrices

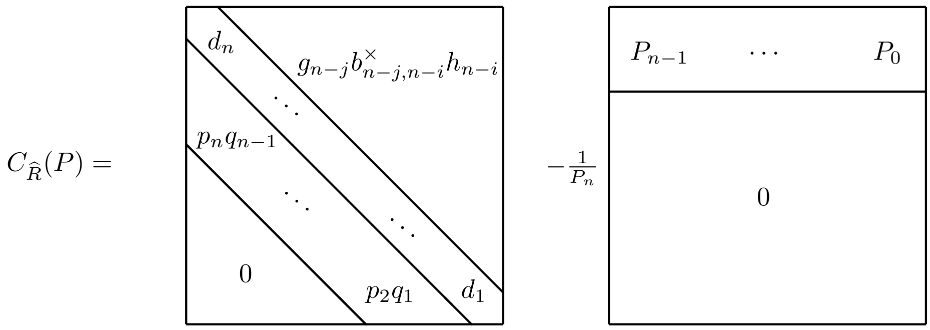
Let $R = \{r_k(x)\}_{k=0}^n$ be a system of (H,m)–quasiseparable polynomials, and $\{x_k\}_{k=1}^n$ a set of distinct nodes. Decomposing the master polynomial into the R basis as

$$\prod_{k=1}^{n}(x-x_k) =: P(x) = P_0\cdot r_0(x) + P_1\cdot r_1(x) + \ldots + P_{n-1}\cdot r_{n-1}(x) + P_n\cdot r_n(x)$$

yields the coefficients $P_0, P_1, \ldots, P_n$, and we have

$$C_R(P) = \begin{bmatrix} d_1 & & g_i b_{ij}^{\times} h_j \\ p_2 q_1 & \ddots & \\ & \ddots & \ddots \\ 0 & p_n q_{n-1} & d_n \end{bmatrix} - \frac{1}{P_n}\begin{bmatrix} & & P_0 \\ & 0 & \vdots \\ & & P_{n-1} \end{bmatrix} \tag{4.1}$$

Applying (2.8) gives us the confederate matrix for the associated polynomials as



(4.2)

From this last equation we can see that the n-term recurrence relations satisfied by the associated polynomials $\widehat{R}$ are given by

$$\widehat{r}_k(x) = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k}\Bigg[\underbrace{(x-\widehat{d}_k)\widehat{r}_{k-1}(x) - \sum_{j=0}^{k-2}\Big(\widehat{g}_{j+1}\widehat{b}^{\times}_{j+1,k}\widehat{h}_k\widehat{r}_j(x)\Big)}_{\text{typical term as in (3.2)}} - \underbrace{\frac{P_{n-k}}{P_n}\widehat{r}_0(x)}_{\text{perturbation term}}\Bigg] \tag{4.3}$$

where, in order to simplify the formulas, we introduce the notation

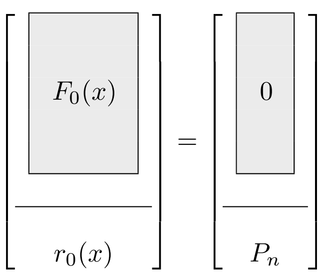
$$\widehat{p}_k = q_{n-k+1}, \quad \widehat{q}_k = p_{n-k+1}, \quad \widehat{d}_k = d_{n-k+1}, \quad \widehat{g}_k = h^T_{n-k+1}, \quad \widehat{b}_k = b^T_{n-k+1}, \quad \widehat{h}_k = g^T_{n-k+1}. \tag{4.4}$$

We will see in a moment that the *nonzero top row* of the second matrix in (4.2) introduces perturbation terms into all formulas that we derive. These perturbation terms are the cause of the increase in quasiseparable rank by at most one.

4.2. Perturbed [EGO05]–type recurrence relations

The previous section showed how n–term recurrence relations for a system of (H,m)–quasiseparable polynomials change after a rank one perturbation of the first row of the corresponding confederate matrix. This small increase in quasiseparable rank allows construction of the desired sparse recurrence relations for associated polynomials, as presented in the next theorem.

Theorem 4.1 (Perturbed [EGO05]–type recurrence relations). *Let $R = \{r_0(x), \ldots, r_{n-1}(x), P(x)\}$ be a system of (H,m)–quasiseparable polynomials corresponding to a (H,m)–quasiseparable matrix of size $n \times n$ with generators $\{p_k, q_k, d_k, g_k, b_k, h_k\}$ as in Definition 3.1, with the convention that $q_n = 0, b_n = 0$. Then the system of polynomials $\widehat{R}$ associated with R satisfy the recurrence relations*



(4.5)

$$\left[\begin{array}{c} \widehat{F}_k(x) \\ \hline \widehat{r}_k(x) \end{array}\right] = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k} \underbrace{\left[\begin{array}{c|c} \widehat{p}_k\widehat{q}_k \widehat{b}_k^T & -\widehat{q}_k \widehat{g}_k^T \\ \hline \widehat{p}_k \widehat{h}_k^T & x - \widehat{d}_k \end{array}\right] \left[\begin{array}{c} \widehat{F}_{k-1}(x) \\ \hline \widehat{r}_{k-1}(x) \end{array}\right]}_{\text{typical terms}} + \underbrace{\frac{1}{\widehat{p}_{k+1}\widehat{q}_k} \left[\begin{array}{c} 0 \\ \hline P_{n-k} \end{array}\right]}_{perturbationterm} \tag{4.6}$$

with the vector of auxiliary polynomials $\widehat{F}_k(x)$, and the coefficients $P_k, k = 0, \dots, n$ are as defined in (1.3).

Proof. The recurrence relations (4.6) define a system of polynomials which satisfy the n-term recurrence relations

$$\widehat{r}_k(x) = (\alpha_k x - a_{k-1,k}) \cdot \widehat{r}_{k-1}(x) - a_{k-2,k} \cdot \widehat{r}_{k-2}(x) - \dots - a_{0,k} \cdot \widehat{r}_0(x) \tag{4.7}$$

for some coefficients $\alpha_k, a_{k-1,k}, \dots, a_{0,k}$. The proof is presented by showing that these n-term recurrence relations in fact coincide exactly with (4.3), so these coefficients coincide with those of the n-term recurrence relations of the associated polynomials $\widehat{R}$; that is,

$$\alpha_k = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k}, \quad a_{k-1,k} = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k}\widehat{d}_k, \quad a_{0,k} = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k}\left(\widehat{g}_1 \widehat{b}^{\times}_{1,k} \widehat{h}_k - \frac{P_{n-k}}{P_n}\right) \tag{4.8}$$

$$a_{j,k} = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k} \widehat{g}_{j+1} \widehat{b}^{\times}_{j+1,k} \widehat{h}_k, \quad j = 1, \dots, k-2$$

Using relations for $\widehat{r}_k(x)$ and $\widehat{F}_{k-1}(x)$ from (4.6), we have

$$\widehat{r}_k(x) = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k}\left[(x - \widehat{d}_k)\widehat{r}_{k-1}(x) - \widehat{g}_{k-1}\widehat{h}_k \widehat{r}_{k-2}(x) + \widehat{p}_{k-1}\widehat{h}_k^T \widehat{b}_{k-1}^T \widehat{F}_{k-2}(x) + \frac{P_{n-k}}{P_n}\widehat{r}_0(x)\right]. \tag{4.9}$$

Notice that again using (4.6) to eliminate $\widehat{F}_{k-2}(x)$ from the equation (4.9) will result in an expression for $\widehat{r}_k(x)$ in terms of $\widehat{r}_{k-1}(x), \widehat{r}_{k-2}(x), \widehat{r}_{k-3}(x), \widehat{F}_{k-3}(x)$, and $\widehat{r}_0(x)$ without modifying the coefficients of $\widehat{r}_{k-1}(x), \widehat{r}_{k-2}(x)$, or $\widehat{r}_0(x)$. Again applying (4.6) to eliminate $\widehat{F}_{k-3}(x)$ results in an expression in terms of $\widehat{r}_{k-1}(x), \widehat{r}_{k-2}(x), \widehat{r}_{k-3}(x), \widehat{r}_{k-4}(x), \widehat{F}_{k-4}(x)$, and $\widehat{r}_0(x)$ without modifying the coefficients of $\widehat{r}_{k-1}(x), \widehat{r}_{k-2}(x)$, $\widehat{r}_{k-3}(x)$, or $\widehat{r}_0(x)$. Continuing in this way, the n-term recurrence relations of the form (4.7) are obtained without modifying the coefficients of the previous ones.

Suppose that for some $0 < j < k-1$ the expression for $\widehat{r}_k(x)$ is of the form

$$\widehat{r}_k(x) = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k}\Big[(x - \widehat{d}_k)\widehat{r}_{k-1}(x) - \widehat{g}_{k-1}\widehat{h}_k \widehat{r}_{k-2}(x) - \cdots$$
$$- \widehat{g}_{j+1}\widehat{b}^{\times}_{j+1,k}\widehat{h}_k \widehat{r}_j(x) + \widehat{p}_{j+1}\widehat{h}_k^T \left(\widehat{b}^{\times}_{j,k}\right)^T \widehat{F}_j(x) + \frac{P_{n-k}}{P_n}\widehat{r}_0(x)\Big]. \tag{4.10}$$

Using (4.6) for $\widehat{F}_j(x)$ gives the relation

$$\widehat{F}_j(x) = \frac{1}{\widehat{p}_{j+1}\widehat{q}_j}\left(\widehat{p}_j \widehat{q}_j \widehat{b}_j^T \widehat{F}_{j-1}(x) - \widehat{q}_j \widehat{g}_j^T \widehat{r}_{j-1}(x)\right) \tag{4.11}$$

Inserting (4.11) into (4.10) gives

$$\begin{aligned} \widehat{r}_k(x) \quad = \quad & \frac{1}{\widehat{p}_{k+1}\widehat{q}_k}\Big[(x - \widehat{d}_k)\widehat{r}_{k-1}(x) - \widehat{g}_{k-1}\widehat{h}_k \widehat{r}_{k-2}(x) - \cdots - \widehat{g}_j \widehat{b}^{\times}_{j,k}\widehat{h}_k \widehat{r}_{j-1}(x) \\ & + \widehat{p}_j \widehat{h}_k^T \left(\widehat{b}^{\times}_{j-1,k}\right)^T \widehat{F}_{j-1}(x) + \frac{P_{n-k}}{P_n}\widehat{r}_0(x)\Big]. \end{aligned} \tag{4.12}$$

Therefore since (4.9) is the case of (4.10) for $j = k-2$, (4.10) is true for each $j = k-2, k-3, \dots, 0$, and for $j = 0$, using the fact that $\widehat{F}_0 = 0$ we have

$$\widehat{r}_k(x) \quad = \quad \frac{1}{\widehat{p}_{k+1}\widehat{q}_k}\left[(x - \widehat{d}_k)\widehat{r}_{k-1}(x) - \widehat{g}_{k-1}\widehat{h}_k \widehat{r}_{k-2}(x) - \cdots - \widehat{g}_1 \widehat{b}^{\times}_{1,k}\widehat{h}_k \widehat{r}_0(x) + \frac{P_{n-k}}{P_n}\widehat{r}_0(x)\right] \tag{4.13}$$

Since these coefficients coincide with those in (4.8) that are satisfied by the associated polynomials, the polynomials given by (4.6) must coincide with the associated polynomials. This proves the theorem. □

4.3. Known special cases of these more general recurrence relations

In this section, recurrence relations valid for the class of polynomials associated with (H,m)–quasiseparable polynomials were derived. Such are needed to provide a Traub–like algorithm. We next demonstrate that these more general recurrence relations reduce as expected in the classical cases. That is, since monomials and real orthogonal polynomials are themselves (H,m)–quasiseparable, the above formulas are valid for those classes as well, and furthermore, the special cases of these formulas are, in fact, the classical formulas for these cases.

Example 4.2 (Classical Traub case: monomials and the Horner polynomials). As shown earlier, the well known companion matrix (2.5) results when the polynomial system R is simply a system of monomials. By choosing the generators $p_k = 1, q_k = 1, d_k = 0, g_k = 1, b_k = 1$, and $h_k = 0$, the matrix (4.1) reduces to (2.5), and also (4.2) reduces to the confederate matrix for the Horner polynomials (2.7). In this special case, the perturbed three-term recurrence relations of Theorem 4.1 become

$$\widehat{r}_0(x) = P_n, \quad \widehat{r}_k(x) = x\widehat{r}_{k-1}(x) + P_{n-k}, \tag{4.14}$$

after eliminating the auxiliary polynomials present, coinciding with the known recurrence relations for the Horner polynomials, used in the evaluation of the polynomial

$$P(x) = P_0 + P_1 x + \cdots + P_{n-1}x^{n-1} + P_n x^n. \tag{4.15}$$

Example 4.3 (Calvetti–Reichel case: Real orthogonal polynomials and the Clenshaw rule). Consider the almost tridiagonal confederate matrix

$$C_R(P) = \begin{bmatrix} d_1 & h_2 & 0 & \cdots & 0 & -P_0/P_n \\ q_1 & d_2 & h_3 & \ddots & \vdots & -P_1/P_n \\ 0 & q_2 & d_3 & h_4 & 0 & \vdots \\ 0 & 0 & q_3 & d_4 & \ddots & -P_{n-3}/P_n \\ \vdots & \ddots & \ddots & \ddots & \ddots & h_n - P_{n-2}/P_n \\ 0 & \cdots & 0 & 0 & q_{n-1} & d_n - P_{n-1}/P_n \end{bmatrix}. \tag{4.16}$$

The corresponding system of polynomials R satisfy three-term recurrence relations; for instance, the highlighted column implies

$$r_3(x) = \frac{1}{q_3}(x - d_3)r_2(x) - \frac{h_3}{q_3}r_1(x) \tag{4.17}$$

by the definition of the confederate matrix. Thus, confederate matrices of this form correspond to systems of polynomials satisfying three-term recurrence relations, or systems of polynomials orthogonal on a real interval, and the polynomial $P(x)$. Such confederate matrices can be seen as special cases of our general class by choosing scalar generators, with $p_k = 1, b_k = 0$, and $g_k = 1$, and in this case the matrix (4.1) reduces to (4.16).

With these choices of generators, applying Theorem 4.1 and eliminating the auxiliary polynomials yields the recurrence relations

$$\widehat{r}_k(x) = \frac{1}{q_{n-k}}(x - d_{n-k})\widehat{r}_{k-1}(x) - \frac{q_{n-k+1}}{q_{n-k}}h_{n-k+1}\widehat{r}_{k-2}(x) + \frac{1}{q_{n-k}}P_{n-k} \tag{4.18}$$

which coincides with the Clenshaw rule for evaluating a polynomial in a real orthogonal basis, i.e. of the form

$$P(x) = P_0 r_0(x) + P_1 r_1(x) + \cdots + P_{n-1}r_{n-1}(x) + P_n r_n(x). \tag{4.19}$$

By the discussion of pertransposition in Section 2, recurrence relations for the system of polynomials associated with real orthogonal polynomials can be found by considering the confederate matrix

$$C_R(P) = \begin{bmatrix} d_n - P_{n-1}/P_n & h_n - P_{n-2}/P_n & -P_{n-3}/P_n & \cdots & -P_1/P_n & -P_0/P_n \\ q_{n-1} & d_{n-1} & h_{n-1} & \ddots & \vdots & \\ 0 & q_{n-2} & d_{n-2} & h_{n-2} & 0 & \vdots \\ 0 & 0 & q_{n-3} & d_{n-3} & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & h_2 \\ 0 & \cdots & 0 & 0 & q_1 & d_1 \end{bmatrix}. \tag{4.20}$$

obtained by pertransposition of (4.16). Note that the highlighted column corresponds to the full recurrence relation

$$\widehat{r}_3(x) = \frac{1}{q_{n-3}}(x - d_{n-2})\widehat{r}_2(x) - \frac{h_{n-1}}{q_{n-3}}\widehat{r}_1(x) + \frac{1}{q_{n-3}}\frac{P_{n-3}}{P_n}\widehat{r}_0(x) \tag{4.21}$$

Thus our formula generalizes both the Clenshaw rule and the algorithms designed for inversion of three-term-Vandermonde matrices in [7] and [14].

5. Computing the coefficients of the master polynomial

Note that in order to use the recurrence relations of the previous section it is necessary to decompose the master polynomial $P(x)$ into the R basis; that is, the coefficients P_k as in (1.3) must be computed. To this end, an efficient method of calculating these coefficients follows.

It is easily seen that the last polynomial $r_n(x)$ in the system R does not affect the resulting confederate matrix $C_R(P)$. Thus, if $\bar{R} = \{r_0(x), \ldots, r_{n-1}(x), xr_{n-1}(x)\}$, we have $C_R(P) = C_{\bar{R}}(P)$. Decomposing the polynomial $P(x)$ into the $\bar{R}$ basis can be done recursively by setting $r_n^{(0)}(x) = 1$ and then for $k = 0, \ldots, n-1$ updating $r_n^{(k+1)}(x) = (x - x_{k+1}) \cdot r_n^{(k)}(x)$. The following lemma gives this procedure, and is from [23].

Lemma 5.1 ([23]). *Let $R = \{r_0(x), \ldots, r_n(x)\}$ be given by (1.2), and $f(x) = \sum_{i=1}^{k} a_i \cdot r_i(x)$, where $k < n-1$. Then the coefficients of $x \cdot f(x) = \sum_{i=1}^{k+1} b_i \cdot r_i(x)$ can be computed by*

$$\begin{bmatrix} b_0 \\ \vdots \\ b_k \\ b_{k+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \left[\begin{array}{cccc|c} \multicolumn{4}{c|}{C_R(r_n)} & 0 \\ \hline 0 & \cdots & 0 & \frac{1}{\alpha_n} & 0 \end{array}\right] \begin{bmatrix} a_0 \\ \vdots \\ a_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.1}$$

Proof. It can be easily checked that

$$x \cdot \begin{bmatrix} r_0(x) & r_1(x) & \cdots & r_n(x) \end{bmatrix} - \begin{bmatrix} r_0(x) & r_1(x) & \cdots & r_n(x) \end{bmatrix} \cdot \left[\begin{array}{cccc|c} \multicolumn{4}{c|}{C_R(r_n)} & 0 \\ \hline 0 & \cdots & 0 & \frac{1}{\alpha_n} & 0 \end{array}\right]$$
$$= \begin{bmatrix} 0 & \cdots & 0 & x \cdot r_n(x) \end{bmatrix}.$$

Multiplying the latter equation by the column of the coefficients we obtain (5.1). □

This lemma suggests the following algorithm for computing coefficients $\{P_0, P_1, \ldots, P_{n-1}, P_n\}$ in

$$\prod_{k=1}^{n}(x - x_k) = P_0 r_0(x) + P_1 r_1(x) + \cdots + P_{n-1} r_{n-1}(x) + P_n r_n(x). \tag{5.2}$$

of the master polynomial.

Algorithm 5.2. *[Coefficients of the master polynomial in the R basis]*
Cost: *$\mathcal{O}(n \times m(n))$, where $m(n)$ is the cost of multiplication of an $n \times n$ quasiseparable matrix by a vector.*
Input: *A quasiseparable confederate matrix $C_R(r_n)$ and n nodes $x = (x_1, x_2, \ldots, x_n)$.*

1. *Set* $\begin{bmatrix} P_0^{(0)} & \cdots & P_{n-1}^{(0)} & P_n^{(0)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$
2. *For* $k = 1 : n$,

$$\begin{bmatrix} P_0^{(k)} \\ \vdots \\ P_{n-1}^{(k)} \\ P_n^{(k)} \end{bmatrix} = \left(\left[\begin{array}{c|c} C_{\bar{R}}(x \cdot r_{n-1}(x)) & 0 \\ \hline 0 \ \cdots \ 0 \ 1 & 0 \end{array} \right] - x_k \cdot I \right) \cdot \begin{bmatrix} P_0^{(k-1)} \\ \vdots \\ P_{n-1}^{(k-1)} \\ P_n^{(k-1)} \end{bmatrix}$$

where $\bar{R} = \{r_0(x), \ldots, r_{n-1}(x), xr_{n-1}(x)\}$.

3. *Take* $\begin{bmatrix} P_0 & \cdots & P_{n-1} & P_n \end{bmatrix} = \begin{bmatrix} P_0^{(n)} & \cdots & P_{n-1}^{(n)} & P_n^{(n)} \end{bmatrix}$

Output: *Coefficients* $\{P_0, P_1, \ldots, P_{n-1}, P_n\}$ *such that* (5.2) *is satisfied.*

It is clear that the computational burden in implementing this algorithm is in multiplication of the matrix $C_{\bar{R}}(r_n)$ by the vector of coefficients. The cost of each such step is $\mathcal{O}(m(n))$, where $m(n)$ is the cost of multiplication of an $n \times n$ quasiseparable matrix by a vector, thus the cost of computing the n coefficients is $\mathcal{O}(n \times m(n))$. Using a fast $\mathcal{O}(n)$ algorithm for multiplication of a quasiseparable matrix by a vector first derived in [9] (or its matrix interpretation of [4]), the cost of this algorithm is $\mathcal{O}(n^2)$.

6. The overall Traub–like algorithm

6.1. Quasiseparable generator input

The main algorithm of this section is the Traub–like algorithm that outputs the inverse of a (H, m)–quasiseparable–Vandermonde matrix. It takes as input the generators $\{p_k, q_k, d_k, g_k, b_k, h_k\}$ of the (H, m)–quasiseparable confederate matrix corresponding to the system of polynomials R.

In this algorithm we will make use of MATLAB notations; for instance $V_{\widehat{R}}(i : j, k : l)$ will refer to the block of $V_{\widehat{R}}(x)$ consisting of rows i through j and columns k through l. For each node x_k we have a vector of auxiliary polynomials $F_{\widehat{R}}(x_k)$. Let us compose a matrix of such vectors $[F_{\widehat{R}}(x_1)| \cdots |F_{\widehat{R}}(x_n)]$ and denote it as $\widehat{F_k}$ on each step.

Algorithm 6.1. *[Traub–like inversion algorithm]*
Cost: $\mathcal{O}(n^2)$ *operations.*
Input: *Generators* $\{p_k, q_k, d_k, g_k, b_k, h_k\}$ *of a quasiseparable confederate matrix corresponding to a system of polynomials* R *and* n *nodes* $x = (x_1, x_2, \ldots, x_n)$.

1. *Compute the entries of diag*$(c_1, \ldots, c_n)$ *via* (2.2)*:* $c_i = \prod_{\substack{k=1 \\ k \neq i}}^{n} (x_k - x_i)^{-1}$.
2. *Compute the coefficients* $\{P_0, \ldots, P_n\}$ *of the master polynomial* $P(x)$ *as in* (1.3) *via Algorithm* 5.2.
3. *Evaluate the* n *polynomials of* $\widehat{R}$ *specified via* (2.8) *at the* n *nodes* x_k *to form* $V_{\widehat{R}}(x)$. *Theorems* 4.1 *provides an algorithm for this.*

   (a) *Set* $V_{\widehat{R}}(:, 1) = P_n \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$, $\quad \widehat{F_1} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}$.

   (b) *For* $k = 1 : n-1$, *compute*

$$V_{\widehat{R}}(:, k+1) = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k} \left(\widehat{p}_k \widehat{F_k} \widehat{h}_k + \left(\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} - \widehat{d}_k \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right) V_{\widehat{R}}(:, k) + P_{n-k} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right) \right)$$

   and

$$F_{\widehat{R}}(:, k+1) = \frac{1}{\widehat{p}_{k+1}\widehat{q}_k} \left(\widehat{p}_k \widehat{q}_k \widehat{b}_k^T \widehat{F_k} - \widehat{q}_k \widehat{g}_k^T V_{\widehat{R}}(:, k)^T \right)$$

   Note: The product of two column vectors is understood to be componentwise.

4. *Compute the inverse $V_R(x)^{-1}$ via (2.1):*

$$V_R(x)^{-1} = \tilde{I} \cdot V_{\widehat{R}}^T(x) \cdot \operatorname{diag}(c_1, \dots, c_n)$$

Output: *Entries of $V_R(x)^{-1}$, the inverse of the polynomial-Vandermonde matrix.*

6.2. Recurrence relation coefficient input

The previous section provides the Traub–like algorithm, which takes as input the generators of the (H,m)–quasiseparable polynomials involved in forming the quasiseparable–Vandermonde matrix. However, as in the motivating cases of real orthogonal polynomials and Szegö polynomials, problems may be stated in terms of the coefficients of the involved recurrence relations instead of in terms of generators.

In this section, we present a result allowing conversion from the language of recurrence relation coefficients to that of quasiseparable generators. Applying this conversion as a preprocessor, the algorithm of the previous section can then be used for problems stated in terms of recurrence relation coefficients.

Theorem 6.2 (Recurrence relations coefficients $\Rightarrow$ quasiseparable generators). *Let $R = \{r_k(x)\}_{k=0}^n$ be a system of polynomials satisfying the [EGO05]–type two-term recurrence relations (3.4):*

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

Then the (H,m)–quasiseparable matrix

$$C_R(r_n) = \begin{bmatrix} -\frac{\theta_1}{\delta_1} & -(\frac{1}{\delta_2})\gamma_2\beta_1 & -\frac{1}{\delta_3}\gamma_3\alpha_2\beta_1 & -\frac{1}{\delta_4}\gamma_4\alpha_3\alpha_2\beta_1 & \cdots & -\frac{1}{\delta_n}\gamma_n\alpha_{n-1}\alpha_{n-2}\cdots\alpha_3\alpha_2\beta_1 \\ \frac{1}{\delta_1} & -\frac{\theta_2}{\delta_2} & -\frac{1}{\delta_3}\gamma_3\beta_2 & -\frac{1}{\delta_4}\gamma_4\alpha_3\beta_2 & \cdots & -\frac{1}{\delta_n}\gamma_n\alpha_{n-1}\alpha_{n-2}\cdots\alpha_3\beta_2 \\ 0 & \frac{1}{\delta_2} & -\frac{\theta_3}{\delta_3} & -\frac{1}{\delta_4}\gamma_4\beta_3 & \ddots & -\frac{1}{\delta_n}\gamma_n\alpha_{n-1}\cdots\alpha_4\beta_3 \\ 0 & 0 & \frac{1}{\delta_3} & -\frac{\theta_4}{\delta_4} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & -\frac{1}{\delta_n}\gamma_n\beta_{n-1} \\ 0 & \cdots & 0 & 0 & \frac{1}{\delta_{n-1}} & -\frac{\theta_n}{\delta_n} \end{bmatrix} \tag{6.1}$$

with generators

$$d_k = -\frac{\theta_k}{\delta_k}, \quad (k=1,\dots,n), \qquad p_{k+1}q_k = \frac{1}{\delta_k}, \quad (k=1,\dots,n-1),$$

$$\boxed{g_k} = \boxed{\beta_k^T}, \quad (k=1,\dots,n-1),$$

$$\boxed{b_k} = \boxed{\alpha_k^T}, \quad (k=2,\dots,n-1), \qquad \boxed{h_k} = -\frac{1}{\delta_k}\boxed{\gamma_k^T}, \quad (k=2,\dots,n)$$

is a confederate matrix for the system of polynomials R.

Proof. Inserting the specified choice of generators into the general n-term recurrence relations (3.2), we arrive at

$$\begin{aligned} r_k(x) &= (\delta_k x + \theta_k) r_{k-1}(x) + \gamma_k\beta_{k-1} r_{k-2}(x) + \gamma_k\alpha_{k-1}\beta_{k-2} r_{k-3}(x) \\ &\quad + \gamma_k\alpha_{k-1}\alpha_{k-2}\beta_{k-3} r_{k-4}(x) + \cdots + \gamma_k\alpha_{k-1}\cdots\alpha_2\beta_1 r_0(x) \end{aligned} \tag{6.2}$$

It suffices to show that the polynomial system satisfying the two-term recurrence relations also satisfies these n-term recurrence relations. Beginning with

$$r_k(x) = (\delta_k x + \theta_k) r_{k-1}(x) + \gamma_k F_{k-1}(x) \tag{6.3}$$

and using the relation $F_{k-1}(x) = \alpha_{k-1}F_{k-2}(x) + \beta_{k-1}r_{k-2}(x)$, (6.3) becomes

$$r_k(x) = (\delta_k x + \theta_k) r_{k-1}(x) + \gamma_k\beta_{k-1} r_{k-2}(x) + \gamma_k\alpha_{k-1} F_{k-2}(x)$$

and continuing this procedure to obtain n-term recurrence relations. It can easily be checked that this procedure yields exactly (6.2). □

7. Numerical Experiments

The numerical properties of the Traub algorithm and its generalizations (that are the special cases of the algorithm proposed in this paper) were studied by different authors. It was noticed in [16] that a modification of the classical Traub algorithm of [28] can yield high accuracy in certain cases if the algorithm is preceded with the *Leja ordering* of the nodes; that is, ordering such that

$$|x_1| = \max_{1\leqslant i\leqslant n} |x_i|, \qquad \prod_{j=1}^{k-1} |x_k - x_j| = \max_{k\leqslant i\leqslant n} \prod_{j=1}^{k-1} |x_i - x_j|, \quad k = 2, \dots, n-1$$

(see [26], [19], [24]) It was noticed in [16] that the same is true for Chebyshev-Vandermonde matrices.

No error analysis was done, but the conclusions of the above authors was that in many cases the Traub algorithm and its extensions can yield much better accuracy than Gaussian elimination, even for very ill-conditioned matrices.

We made our preliminary experiments with the proposed Traub–like algorithm, and our conclusions for the most general case are consistent with the experience of our colleagues made for special cases. The results of experiments with the proposed algorithm yields better accuracy than Gaussian elimination. However, our experiments need to be done for different special cases of (H, m)–quasiseparable polynomials.

The algorithm was implemented in *C* using *Lapack* for all supplementary matrix computations (such as matrix multiplication GEMM). For Gaussian elimination we used the *Lapack* subroutine GESV. To estimate the accuracy of all of the above algorithms we took the output of new Traub–like algorithm in double precision $V_R(x)^{-1}$ as the exact solution.

We compare the forward accuracy of the inverse computed by the algorithm in single precision $\widehat{V_R(x)^{-1}}$ with respect to the inverse computed in double precision, defined by

$$e = \frac{\|\widehat{V_R(x)^{-1}} - V_R(x)^{-1}\|_2}{\|V_R(x)^{-1}\|_2} \tag{7.1}$$

where $V_R^s(x)^{-1}$ is the solution computed by each algorithm in single precision. In the tables, New Algorithm denotes the proposed Traub–like algorithm with Leja ordering, and GESV indicates *Lapack*'s inversion subroutine. Finally, cond(V) denotes the condition number of the matrix V computed via the *Lapack* subroutine GESVD.

Experiment 1. (Random choice of generators) In this experiment, the generators we chosen randomly in $(-1, 1)$, and the nodes x_k were selected equidistant on $(-1, 1)$ via the formula

$$x_k = -1 + 2\left(\frac{k}{n-1}\right), \quad k = 0, 1, \dots, n-1$$

We test the accuracy of the inversion algorithm for various sizes n and quasiseparable ranks m of matrices generated in this way. Some results are tabulated in Table 4.

Notice that the performance of the proposed inversion algorithm is an improvement over that of *Lapack*'s standard inversion subroutine GESV in this specific case. And in almost all cases relative errors are around e-7, which means that all digits of the errors in single precision coincide with corresponding digits in double precision. There are occasional examples in which the proposed algorithm can lose several decimal digits, but it still outperforms Gaussian elimination.

Experiment 2. (l-recurrent polynomials) In this experiment we consider l-recurrent polynomials

$$r_k(x) = (\alpha_k x - a_{k-1,k}) \cdot r_{k-1}(x) - a_{k-2,k} \cdot r_{k-2}(x) - \dots - a_{k-(l-1),k} \cdot r_{k-(l-1)}(x) \tag{7.2}$$

by choosing coefficients of (7.2) randomly in $(-1, 1)$, and the nodes x_k equidistant on $(-1, 1)$.

We test the accuracy of the inversion algorithm for various sizes n and number of terms l. We remind the reader that quasiseparable rank of polynomials given by (7.2) is $l-2$. Table 4 presents some results generated in this way.

Table 4. Random generators on (-1,1). Equidistant nodes on (-1,1).

n	m	cond(V)	GESV	New Algorithm
10	1	1.8e+007	1.5e-006	1.4e-006
	2	5.6e+007	4.6e-005	5.4e-007
20	1	2.2e+020	5.9e-001	5.0e-007
	2	1.6e+019	2.6e+000	1.9e-007
	3	1.0e+021	5.6e-002	2.3e-006
	4	6.1e+020	2.8e+000	5.5e-006
30	1	7.2e+029	1.2e+000	2.6e-006
	2	3.4e+025	9.2e-001	2.7e-006
	3	2.9e+029	1.0e+000	2.0e-006
	4	7.5e+026	1.0e+000	1.5e-006
	5	5.0e+024	1.0e+000	1.4e-006
	6	2.5e+026	1.0e+000	5.1e-007
40	1	2.1e+034	1.0e+000	1.6e-005
	2	3.3e+033	1.0e+000	1.9e-006
	3	4.1e+029	1.0e+000	1.1e-003
	4	4.5e+028	1.0e+000	3.5e-007
	5	1.2e+031	1.0e+000	7.2e-007
	6	3.5e+032	1.0e+000	3.3e-006
	7	6.0e+027	1.0e+000	1.7e-004
	8	7.8e+031	1.0e+000	5.0e-007
50	1	1.5e+039	1.0e+000	3.9e-007
	2	3.7e+038	1.0e+000	7.6e-001
	3	2.6e+041	1.0e+000	4.5e-004
	4	2.0e+037	1.0e+000	3.8e-001
	5	1.7e+037	1.0e+000	5.7e-007
	6	8.0e+038	1.0e+000	9.2e-005
	7	1.7e+038	1.0e+000	9.3e-007
	8	7.5e+036	1.0e+000	4.7e-007

8. Conclusions

In this paper we extend the previous work in the area of fast Traub–like inversion algorithms to the general class of (H,m)–quasiseparable–Vandermonde matrices. This generalizes results for Vandermonde, three–term–Vandermonde, Szegö–Vandermonde, and $(H,1)$–quasiseparable–Vandermonde matrices. Exploiting the quasiseparable structure yields sparse recurrence relations which allow the desired computational speedup, resulting in a fast $\mathcal{O}(n^2)$ algorithm as opposed to Gaussian elimination, which requires $\mathcal{O}(n^3)$ operations. Finally, some numerical experiments were presented that indicate that, under some circumstances, the resulting algorithm can give better performance than Gaussian elimination.

References

[1] M.Bakonyi and T.Constantinescu, *Schur's algorithm and several applications*, in Pitman Research Notes in Mathematics Series, vol. 61, Longman Scientific and Technical, Harlow, 1992.

[2] A. Björck and V. Pereyra, *Solution of Vandermonde Systems of Equations*, Math. Comp., **24** (1970), 893-903.

[3] T.Bella, Y.Eidelman, I.Gohberg, I.Koltracht and V.Olshevsky, *A Björck–Pereyra–type algorithm for Szegö-Vandermonde matrices based on properties of unitary Hessenberg matrices*, Linear Algebra and Applications, Volume 420, Issues 2-3 pp. 634-647, 2007.

[4] T.Bella, Y.Eidelman, I.Gohberg, I.Koltracht and V.Olshevsky *A fast Bjorck–Pereyra like algorithm for solving Hessenberg-quasiseparable-Vandermonde systems*, submitted to SIAM Journal of Matrix Analysis (SIMAX), 2007.

Table 5. l-recurrent polynomials. Random coefficients on (-1,1).

n	l	cond(V)	inv()	TraubQS(Leja)
10	3	9.5e+004	1.5e-004	1.9e-007
	4	1.2e+006	7.1e-004	3.3e-007
20	3	4.3e+013	1.0e+000	2.8e-007
	4	2.2e+013	1.0e+000	3.4e-007
	5	1.5e+012	1.0e+000	5.1e-007
	6	4.1e+011	1.0e+000	2.2e-007
30	3	4.7e+016	1.0e+000	4.2e-007
	4	1.8e+016	1.0e+000	3.2e-007
	5	3.0e+018	1.0e+000	4.4e-007
	6	7.3e+016	1.0e+000	4.1e-007
	7	1.2e+017	1.0e+000	5.1e-007
	8	3.6e+017	1.0e+000	2.8e-007
40	3	8.9e+017	1.0e+000	4.8e-007
	4	1.2e+020	1.0e+000	6.5e-007
	5	2.3e+018	1.0e+000	8.3e-007
	6	2.2e+021	1.0e+000	4.5e-007
	7	2.4e+020	1.0e+000	6.8e-007
	8	1.8e+018	1.0e+000	9.7e-007
	9	8.9e+019	1.0e+000	1.1e-006
	10	8.6e+020	1.0e+000	6.3e-007
50	3	3.6e+018	1.0e+000	2.8e-007
	4	3.1e+019	1.0e+000	2.0e-007
	5	5.3e+019	1.0e+000	6.1e-007
	6	7.8e+019	1.0e+000	4.8e-007
	7	1.8e+020	1.0e+000	1.8e-007
	8	3.6e+019	1.0e+000	4.2e-007
	9	7.0e+019	1.0e+000	5.5e-007
	10	2.2e+020	1.0e+000	8.5e-007
	11	3.9e+021	1.0e+000	1.8e-007
	12	5.3e+020	1.0e+000	5.3e-007

[5] 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.

[6] 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.

[7] Calvetti, D. and Reichel, L. : *Fast inversion of Vandermonde–like matrices involving orthogonal polynomials*, BIT, 1993.

[8] Y. Eidelman and I. Gohberg, *On a new class of structured matrices*, Integral Equations and Operator Theory, **34** (1999), 293-324.

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

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

[11] 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, Volume 405, 1 August 2005, Pages 1-40.

[12] G.Forney, *Concatenated codes*, The M.I.T. Press, 1966, Cambridge.

[13] L.Y.Geronimus, *Polynomials orthogonal on a circle and their applications*, Amer. Math. Translations, **3** p.1-78, 1954 (Russian original 1948).

[14] I.Gohberg and V.Olshevsky, *Fast inversion of Chebyshev-Vandermonde matrices*, Numerische Mathematik, **67, No. 1** (1994), 71-92.

[15] I.Gohberg and V.Olshevsky, *A fast generalized Parker-Traub algorithm for inversion of Vandermonde and related matrices*, Journal of Complexity, **13(2)** (1997), 208-234.
A short version in pp. in *Communications, Computation, Control and Signal Processing: A tribute to Thomas Kailath*, Eds. A.Paulraj, V Roychowdhury and C.Shaper, Kluwer Academic Publishing, 1996, 205-221.

[16] 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.

[17] U.Grenader and G.Szegö, *Toeplitz forms and Applications*, University of California Press, 1958.

[18] W.G.Horner, *A new method of solving numerical equations of all orders by continuous approximation*, Philos. Trans. Roy. Soc. London, (1819), 308-335.

[19] N.J.Higham, *Stability analysis of algorithms for solving confluent Vandermonde–like systems*, SIAM J. Matrix Anal. Appl., **11**(1) (1990), 23–41.

[20] T.Kailath and V.Olshevsky, *Displacement structure approach to polynomial Vandermonde and related matrices*, Linear Algebra and Its Applications, **261** (1997), 49-90.

[21] J.Maroulas and S.Barnett, Polynomials with respect to a general basis. I. Theory, J. of Math. Analysis and Appl., **72** (1979), 177 -194.

[22] 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.

[23] V.Olshevsky, *Associated polynomials, unitary Hessenberg matrices and fast generalized Parker-Traub and Bjorck-Pereyra algorithms for Szego-Vandermonde matrices* invited chapter in the book "Structured Matrices: Recent Developments in Theory and Computation," 67-78, (D.Bini, E. Tyrtyshnikov, P. Yalamov., Eds.), 2001, NOVA Science Publ., USA.

[24] V. Olshevsky, *Pivoting for structured matrices and rational tangential interpolation*, in Fast Algorithms for Structured Matrices: Theory and Applications, CONM/323, p. 1 - 75, AMS publications, May 2003.

[25] F.Parker, *Inverses of Vandermonde matrices*, Amer. Math. Monthly, **71** (1964), 410 - 411.

[26] L.Reichel and G.Opfer, *Chebyshev-Vandermonde systems*, Math. of Comp., **57** (1991), 703-721.

[27] J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, 1992. 277-301.

[28] J. Traub, *Associated polynomials and uniform methods for the solution of linear problems*, SIAM Review, **8, No. 3** (1966), 277-301.

T.Bella* and Y.Eidelman† and I.Gohberg† and V.Olshevsky* and E.Tyrtyshnikov‡ and P.Zhlobich*