A FAST BJÖRCK-PEREYRA-LIKE ALGORITHM FOR SOLVING HESSENBERG-QUASISEPARABLE-VANDERMONDE SYSTEMS

T.BELLA*, Y.EIDELMAN†, I.GOHBERG† , I.KOLTRACHT* , AND V.OLSHEVSKY*

Abstract. In this paper we derive a fast $\mathcal{O}(n^2)$ algorithm for solving linear systems where the coefficient matrix is a polynomial-Vandermonde matrix $V_R(x) = [r_{j-1}(x_i)]$ with polynomials $\{r_k(x)\}$ related to a Hessenberg quasiseparable matrix. The result generalizes the well-known Björck-Pereyra algorithm for classical Vandermonde systems involving monomials. It also generalizes the algorithms of [RO91] for $V_R(x)$ involving Chebyshev polynomials, of [H90] for $V_R(x)$ involving real orthogonal polynomials, and of [BEGKO07] for $V_R(x)$ involving Szegö polynomials. The new algorithm applies to a fairly general class of H-k-q.s.-polynomials (Hessenberg order k quasiseparable) that includes (along with the above mentioned classes of real orthogonal and Szegö polynomials) several other important classes of polynomials. Preliminary numerical experiments are presented comparing the algorithm to standard structure-ignoring methods.

1. Introduction.

1.1. Polynomial-Vandermonde matrices and Björck-Pereyra-type algorithms. The fact that the n^2 entries of Vandermonde matrices $V(x) = \left[\ x_i^{j-1}\ \right]$ are determined by only n parameters $\{x_k\}$ allows the design of fast algorithms. Specifically, an algorithm due to Björck and Pereyra [BP70] can solve the system $V(x)a = f$ in $\mathcal{O}(n^2)$ operations. This is as opposed to the $\mathcal{O}(n^3)$ operations required by Gaussian elimination. Moreover, it was proven in [H87] that in many cases the Björck-Pereyra algorithm is guaranteed to provide a very high accuracy. The reduction in complexity as well as the favorable numerical properties attracted much attention in the numerical linear algebra literature, and the Björck-Pereyra algorithm has been carried over to several other important classes of *polynomial-Vandermonde* matrices

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

defined not only by the nodes $\{x_k\}$ but also by the system of polynomials $\{r_k(x)\}$. Table 1.1 below lists several classes of polynomials for which the Björck-Pereyra-type algorithms[1] are currently available.

Matrices $V_R(x)$	Polynomial systems $R = \{r_k(x)\}$	Fast system solver
Classical Vandermonde	monomials	Björck-Pereyra [BP70]
Chebyshev-Vandermonde	Chebyshev polynomials	Reichel-Opfer [RO91]
Three-Term-Vandermonde	Real orthogonal polynomials	Higham [H90]
Szegö-Vandermonde	Szegö polynomials	[BEGKO07]

TABLE 1.1
Fast $\mathcal{O}(n^2)$ algorithms for several classes of polynomial-Vandermonde matrices.

1.2. Capturing recurrence relations via confederate matrices. To generalize the algorithms in Table 1.1 we will use the concept of a *confederate matrix* introduced in [MB79]. Let polynomials $R = \{r_0(x), r_1(x)\ , \ldots,\ r_n(x)\}$ be specified by the general recurrence n-term relations[2]

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

Define for the polynomial

$$\beta(x) = \beta_0 \cdot r_0(x) + \beta_1 \cdot r_1(x) + \ldots + \beta_{n-1} \cdot r_{n-1}(x) + r_n(x) \tag{1.3}$$

*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.

[1] Along with carrying over the Björck-Pereyra algorithm to various families of polynomials $\{r_k(x)\}$ there are also other (matrix) directions of generalization. E.g., the algorithm had been carried over to the block Vandermonde matrices in [TG81] and to Cauchy matrices in [BKO99].

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

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

$$C_R(\beta) = \underbrace{\begin{bmatrix} \frac{a_{01}}{\alpha_1} & \frac{a_{02}}{\alpha_2} & \frac{a_{03}}{\alpha_3} & \cdots & \frac{a_{0,k}}{\alpha_k} & \cdots & \cdots & \frac{a_{0,n}}{\alpha_n} \\ \frac{1}{\alpha_1} & \frac{a_{12}}{\alpha_2} & \frac{a_{13}}{\alpha_3} & \cdots & \frac{a_{1,k}}{\alpha_k} & \cdots & \cdots & \frac{a_{1,n}}{\alpha_n} \\ 0 & \frac{1}{\alpha_2} & \frac{a_{23}}{\alpha_3} & \cdots & \vdots & \cdots & \cdots & \frac{a_{2,n}}{\alpha_n} \\ 0 & 0 & \frac{1}{\alpha_3} & \ddots & \frac{a_{k-2,k}}{\alpha_k} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{a_{k-1,k}}{\alpha_k} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{1}{\alpha_k} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{a_{n-1,n}}{\alpha_n} \end{bmatrix}}_{C_R(r_n)} - \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \vdots \\ \beta_{n-1} \end{bmatrix} \begin{bmatrix} 0 & 0 & \cdots & 0 & 1/\alpha_n \end{bmatrix}. \tag{1.4}$$

Notice that the coefficients of the recurrence relations for the k^{th} polynomial $r_k(x)$ from (1.2) are contained in the highlighted k^{th} column of $C_R(r_n)$. We refer to [MB79] for many useful properties of the confederate matrix and only recall here that

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

where $[C_R(\beta)]_{k\times k}$ denotes the $k \times k$ leading submatrix of $C_R(\beta)$.

Next in Table 1.2 we list confederate matrices for the polynomial systems[3] of Table 1.1.

Recurrence Relations of R	Confederate matrix $C_R(r_n)$
$r_k(x) = x \cdot r_{k-1}(x)$ Monomials	$\begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & \ddots & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$ Companion matrix
$r_k(x) = (\alpha_k x - \delta_k) r_{k-1}(x) - \gamma_k r_{k-2}(x)$ Real orthogonal polynomials	$\begin{bmatrix} \frac{\delta_1}{\alpha_1} & \frac{\gamma_2}{\alpha_2} & 0 & \cdots & 0 \\ \frac{1}{\alpha_1} & \frac{\delta_2}{\alpha_2} & \ddots & \ddots & \vdots \\ 0 & \frac{1}{\alpha_2} & \ddots & \frac{\gamma_{n-1}}{\alpha_{n-1}} & 0 \\ \vdots & \ddots & \ddots & \frac{\delta_{n-1}}{\alpha_{n-1}} & \frac{\gamma_n}{\alpha_n} \\ 0 & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{\delta_n}{\alpha_n} \end{bmatrix}$ Tridiagonal matrix
$r_k(x) = [\frac{1}{\mu_k} \cdot x + \frac{\rho_k}{\rho_{k-1}} \frac{1}{\mu_k}] r_{k-1}(x) - \frac{\rho_k}{\rho_{k-1}} \frac{\mu_{k-1}}{\mu_k} \cdot x \cdot r_{k-2}(x)$ Szegö polynomials	$\begin{bmatrix} -\rho_1\rho_0^* & \cdots & -\rho_{n-1}\mu_{n-2}\ldots\mu_1\rho_0^* & -\rho_n\mu_{n-1}\ldots\mu_1\rho_0^* \\ \mu_1 & \ddots & -\rho_{n-1}\mu_{n-2}\ldots\mu_2\rho_1^* & -\rho_n\mu_{n-1}\ldots\mu_2\rho_1^* \\ 0 & \ddots & \vdots & \vdots \\ \vdots & & -\rho_{n-1}\rho_{n-2}^* & -\rho_n\mu_{n-1}\rho_{n-2}^* \\ 0 & \cdots & \mu_{n-1} & -\rho_n\rho_{n-1}^* \end{bmatrix}$ Unitary Hessenberg matrix

TABLE 1.2
Polynomial systems and corresponding confederate matrices.

[3] For the monomials and for the 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., [O01] and the references therein.

It turns out that tridiagonal and unitary Hessenberg matrices of Table 1.2 are special cases of the more general class of matrices defined next.

1.3. Hessenberg quasiseparable matrices and polynomials. Recall that a matrix $A = [a_{ij}]$ is called *upper Hessenberg* if all entries below the first subdiagonal are zeros; that is, $a_{ij} = 0$ if $i > j + 1$, and is furthermore *irreducible* if $a_{i+1,i} \neq 0$ for $i = 1, \ldots, n-1$.

DEFINITION 1.1. **(Quasiseparable matrices and polynomials)**

- *A matrix A is called H-m-q.s. (or Hessenberg-m-quasiseparable) if* **(i)** *it is upper Hessenberg, and* **(ii)** $\max(rank A_{12}) = m$ *where the maximum is taken over all symmetric partitions of the form*

$$A = \left[\begin{array}{c|c} * & A_{12} \\ \hline * & * \end{array}\right] \tag{1.5}$$

- *Let $A = [a_{ij}]$ be an irreducible (i.e., $a_{i+1,i} \neq 0$), H-m-q.s. matrix. Then the system of polynomials $\{r_k(x)\}$ related to A via*

$$r_k(x) = \alpha_1 \cdots \alpha_k \det (xI - A)_{(k \times k)} \qquad \textit{(where} \qquad \alpha_i = 1/a_{i+1,i}\textit{)}$$

is called a system of H-m-q.s. polynomials (or Hessenberg-m-quasiseparable polynomials).

We verify next that the class of H-1-q.s. polynomials is wide enough to include monomials, real orthogonal and Szegö polynomials (i.e., all polynomials of Tables 1.1 and 1.2) as special cases. This can be seen by inspecting, for each confederate matrix, its typical submatrix A_{12} from the partition described in (1.5).

1.3.1. Tridiagonal matrices are H-1-q.s. Indeed, if A is tridiagonal, then the submatrix A_{12} has the form

$$A_{12} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \frac{\delta_k}{\alpha_k} & 0 & \cdots & 0 \end{bmatrix}$$

which can easily be observed to have rank one.

1.3.2. Unitary Hessenberg matrices are H-1-q.s. Indeed, if A corresponds to the Szegö polynomials, then the corresponding 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.

1.4. Main result. Fast $\mathcal{O}(n^2)$ Björck-Pereyra-like algorithm for H-1-q.s. Vandermonde matrices. In this paper we derive a generalization of the Björck-Pereyra algorithm that applies to the *general polynomial Vandermonde matrices* $V_R(x)$ involving polynomials $R = \{r_k(x)\}$ satisfying only one restriction: $\deg r_k(x) = k$. However, in this general case the algorithm has the cost of $\mathcal{O}(n^3)$ operations, i.e., it is not fast.

The Björck-Pereyra-like algorithms were known to be fast in a number of special cases, e.g., when the corresponding confederate matrix has some structure. Several well-known structured confederate matrices are listed in Table 1.2, and the corresponding known fast $\mathcal{O}(n^2)$ Björck-Pereyra-like algorithms for them are listed in Table 1.1.

We show that our Björck-Pereyra-like algorithm is fast (also requiring $\mathcal{O}(n^2)$ operations) when the polynomial Vandermonde matrices $V_R(x)$ involves H-m-q.s. polynomials with small m. Since the latter class includes real orthogonal and Szegö polynomials, the new fast algorithm generalizes all the algorithms listed in Table 1.1.

Along with real orthogonal polynomials and Szegö polynomials the class of H-m-q.s. polynomials includes several more families, and the new fast $\mathcal{O}(n^2)$ Björck-Pereyra-like algorithm applies to them as well. As a matter of fact, a particular structure in recurrence relations typically yields a quasiseparable structure of the corresponding confederate matrices $C_R(r_n)$, and this allows the computational speed-up. We describe four such examples (for which the Björck-Pereyra algorithms were not available before) next.

1.5. Several interesting special cases for which the new Björck-Pereyra-type algorithm is fast. In this section we describe several subclasses of H-m-q.s. polynomials such that the corresponding Björck-Pereyra-like algorithm, by virtue of the principal result of this paper, is fast.

1.5.1. m-recurrent polynomials. It is easy to see that if polynomials satisfy m-term recurrence relations

$$r_k(x) = (\alpha_k x - a_{k-1,k}) \cdot r_{k-1}(x) - a_{k-2,k} \cdot r_{k-2}(x) - \ldots - a_{k-(m-1),k} \cdot r_{k-(m-1)}(x) \tag{1.6}$$

then their confederate matrices

$$A = \begin{bmatrix} \frac{a_{0,1}}{\alpha_1} & \cdots & \frac{a_{0,m-1}}{\alpha_m-1} & 0 & \cdots & 0 \\ \frac{1}{\alpha_1} & \frac{a_{1,2}}{\alpha_2} & \cdots & \frac{a_{1,m}}{\alpha_m} & \ddots & \vdots \\ 0 & \frac{1}{\alpha_2} & & & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \frac{a_{n-(m-1),n}}{\alpha_n} \\ \vdots & & \ddots & \frac{1}{\alpha_{n-2}} & & \vdots \\ 0 & \cdots & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{a_{n-1,n}}{\alpha_n} \end{bmatrix} \tag{1.7}$$

are $(1, m-2)$-banded, i.e., they have only one nonzero subdiagonal and $m-2$ nonzero superdiagonals. Clearly, any A_{12} in (1.5) has rank at most $(m-2)$, implying that A is a H-$(m-2)$-q.s. matrix by definition (E.g., tridiagonal matrices are H-1-q.s.). Hence m-recurrent polynomials are H-$(m-2)$-q.s. (E.g., polynomials satisfying three-term recurrence relations are H-1-q.s.).

1.5.2. More general three-term recurrence relations. Consider polynomials satisfying fairly general[4] three-term recurrence relations

$$r_k(x) = (\alpha_k x - \delta_k) \cdot r_{k-1}(x) - (\beta_k x + \gamma_k) \cdot r_{k-2}(x). \tag{1.8}$$

It was observed in [BEGOT06] that the confederate matrix $C_R(r_n)$ of such $\{r_k(x)\}$ has the form

$$\begin{bmatrix} \frac{\delta_1}{\alpha_1} & \frac{\frac{\delta_1}{\alpha_1}\beta_2+\gamma_2}{\alpha_2} & \frac{\frac{\delta_1}{\alpha_1}\beta_2+\gamma_2}{\alpha_2}\left(\frac{\beta_3}{\alpha_3}\right) & \frac{\frac{\delta_1}{\alpha_1}\beta_2+\gamma_2}{\alpha_2}\left(\frac{\beta_3}{\alpha_3}\right)\left(\frac{\beta_4}{\alpha_4}\right) & \cdots & \frac{\frac{\delta_1}{\alpha_1}\beta_2+\gamma_2}{\alpha_2}\left(\frac{\beta_3}{\alpha_3}\right)\left(\frac{\beta_4}{\alpha_4}\right)\cdots\left(\frac{\beta_n}{\alpha_n}\right) \\ \frac{1}{\alpha_1} & \frac{\delta_2}{\alpha_2}+\frac{\beta_2}{\alpha_1\alpha_2} & \frac{\left(\frac{\delta_2}{\alpha_2}+\frac{\beta_2}{\alpha_1\alpha_2}\right)\beta_3+\gamma_3}{\alpha_3} & \frac{\left(\frac{\delta_2}{\alpha_2}+\frac{\beta_2}{\alpha_1\alpha_2}\right)\beta_3+\gamma_3}{\alpha_3}\left(\frac{\beta_4}{\alpha_4}\right) & \cdots & \frac{\left(\frac{\delta_2}{\alpha_2}+\frac{\beta_2}{\alpha_1\alpha_2}\right)\beta_3+\gamma_3}{\alpha_3}\left(\frac{\beta_4}{\alpha_4}\right)\cdots\left(\frac{\beta_n}{\alpha_n}\right) \\ & \frac{1}{\alpha_2} & \frac{\delta_3}{\alpha_3}+\frac{\beta_3}{\alpha_2\alpha_3} & \frac{\left(\frac{\delta_3}{\alpha_3}+\frac{\beta_3}{\alpha_2\alpha_3}\right)\beta_4+\gamma_4}{\alpha_4} & & \\ & & \frac{1}{\alpha_3} & \frac{\delta_4}{\alpha_4}+\frac{\beta_4}{\alpha_3\alpha_4} & & \vdots \\ & & & \frac{1}{\alpha_4} & \ddots & \\ & & & \ddots & \ddots & \frac{\left(\frac{\delta_{n-1}}{\alpha_{n-1}}+\frac{\beta_{n-1}}{\alpha_{n-2}\alpha_{n-1}}\right)\beta_n+\gamma_n}{\alpha_n} \\ & & & & \frac{1}{\alpha_{n-1}} & \frac{\delta_n}{\alpha_n}+\frac{\beta_n}{\alpha_{n-1}\alpha_n} \end{bmatrix}. \tag{1.9}$$

Observe that taking $\beta_k = 0$ for each k, the matrix A of (1.9) reduces to the tridiagonal matrix displayed in Table 1.2. Secondly, inserting the relations

$$\alpha_k = \frac{1}{\mu_k}, \quad \delta_k = -\frac{1}{\mu_k}\frac{\rho_k}{\rho_{k-1}}, \quad \beta_k = \frac{\mu_{k-1}}{\mu_k}\frac{\rho_k}{\rho_{k-1}}, \quad \gamma_k = 0$$

into the matrix A of (1.9) results in the unitary Hessenberg matrix displayed in Table 1.2.

It is easy to verify that the matrix A of (1.9) is irreducible H-1-q.s. Indeed, one can see that in any partition (1.5) the $(k-1)$-st column $A_{12}(:, k-1)$ and the k-th column $A_{12}(:, k)$ of the matrix A_{12} are scalar multiples of each other:

$$A_{12}(:, k) = \frac{\beta_{k+2}}{\alpha_{k+2}} A_{12}(:, k-1).$$

[4] I.e., more general than those of Table 1.2.

E.g., inspect the $2 \times (n-2)$ matrix

$$A_{12} = \begin{bmatrix} \frac{\frac{\delta_1}{\alpha_1}\beta_2+\gamma_2}{\alpha_2}\left(\frac{\beta_3}{\alpha_3}\right) & \frac{\frac{\delta_1}{\alpha_1}\beta_2+\gamma_2}{\alpha_2}\left(\frac{\beta_3}{\alpha_3}\right)\left(\frac{\beta_4}{\alpha_4}\right) & \frac{\frac{\delta_1}{\alpha_1}\beta_2+\gamma_2}{\alpha_2}\left(\frac{\beta_3}{\alpha_3}\right)\left(\frac{\beta_4}{\alpha_4}\right)\left(\frac{\beta_5}{\alpha_5}\right) & \cdots \\ \frac{\left(\frac{\delta_2}{\alpha_2}+\frac{\beta_2}{\alpha_1\alpha_2}\right)\beta_3+\gamma_3}{\alpha_3} & \frac{\left(\frac{\delta_2}{\alpha_2}+\frac{\beta_2}{\alpha_1\alpha_2}\right)\beta_3+\gamma_3}{\alpha_3}\left(\frac{\beta_4}{\alpha_4}\right) & \frac{\left(\frac{\delta_2}{\alpha_2}+\frac{\beta_2}{\alpha_1\alpha_2}\right)\beta_3+\gamma_3}{\alpha_3}\left(\frac{\beta_4}{\alpha_4}\right)\left(\frac{\beta_5}{\alpha_5}\right) & \cdots \end{bmatrix}.$$

Hence, polynomials (1.8) are H-1-q.s.

1.5.3. Szegö-type two-term recurrence relations. Consider polynomials $\{r_k(x)\}$ satisfying general two-term recurrence relations of the Szegö type,

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ (\delta_k x + \theta_k) r_{k-1}(x) \end{bmatrix}. \tag{1.10}$$

Here $\{G_k(x)\}$ are some auxiliary polynomials. The class of polynomials (1.10) includes the classical Szegö polynomials $\{r_k(x)\}$ satisfying[5]

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} 1 & -\rho_k \\ -\rho_k^* & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ x r_{k-1}(x) \end{bmatrix}.$$

It was shown in [BEGOT06] that the confederate matrix $C_R(r_n)$ of $\{r_k(x)\}$ satisfying (1.10) has the form

$$\begin{bmatrix} -\frac{\theta_1+\gamma_1}{\delta_1} & -(\alpha_1-\beta_1\gamma_1)\frac{\gamma_2}{\delta_2} & -(\alpha_1-\beta_1\gamma_1)(\alpha_2-\beta_2\gamma_2)\frac{\gamma_3}{\delta_3} & \cdots & -(\alpha_1-\beta_1\gamma_1)\cdots(\alpha_{n-1}-\beta_{n-1}\gamma_{n-1})\frac{\gamma_n}{\delta_n} \\ \frac{1}{\delta_1} & -\frac{\theta_2+\gamma_2\beta_1}{\delta_2} & -\beta_1(\alpha_2-\beta_2\gamma_2)\frac{\gamma_3}{\delta_3} & \cdots & -\beta_1(\alpha_2-\beta_2\gamma_2)\cdots(\alpha_{n-1}-\beta_{n-1}\gamma_{n-1})\frac{\gamma_n}{\delta_n} \\ 0 & \frac{1}{\delta_2} & -\frac{\theta_3+\gamma_3\beta_2}{\delta_3} & \ddots & -\beta_2(\alpha_3-\beta_3\gamma_3)\cdots(\alpha_{n-1}-\beta_{n-1}\gamma_{n-1})\frac{\gamma_n}{\delta_n} \\ \vdots & \ddots & \ddots \quad \ddots & \ddots & \vdots \\ \vdots & & & \ddots & -\beta_{n-1}(\alpha_{n-1}-\beta_{n-1}\gamma_{n-1})\frac{\gamma_n}{\delta_n} \\ 0 & \cdots & 0 & \frac{1}{\delta_{n-1}} & -\frac{\theta_n+\gamma_n\beta_{n-1}}{\delta_n} \end{bmatrix} \tag{1.11}$$

As in the unitary Hessenberg case, it is easy to see that the rows of A_{12} in are scalar multiples of each other and hence A is H-1-q.s., and the polynomials $\{r_k(x)\}$ in (1.10) are H-1-q.s.

1.5.4. [EGO05]-type two-term recurrence relations. Finally, suppose the polynomials $\{r_k(x)\}$ satisfy the recurrence relations

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & \delta_k x + \theta_k \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ r_{k-1}(x) \end{bmatrix}. \tag{1.12}$$

Again, $\{G_k(x)\}$ are some auxiliary polynomials. It was shown in [BEGOT06] that the confederate matrix $C_R(r_n)$ of such $\{r_k(x)\}$ has the form

$$\begin{bmatrix} -\frac{\theta_1}{\delta_1} & -\beta_1(\frac{\gamma_2}{\delta_2}) & -\beta_1\alpha_2(\frac{\gamma_3}{\delta_3}) & -\beta_1\alpha_2\alpha_3(\frac{\gamma_4}{\delta_4}) & \cdots & -\beta_1\alpha_2\alpha_3\alpha_4\cdots\alpha_{n-1}(\frac{\gamma_n}{\delta_n}) \\ \frac{1}{\delta_1} & -\frac{\theta_2}{\delta_2} & -\beta_2(\frac{\gamma_3}{\delta_3}) & -\beta_2\alpha_3(\frac{\gamma_4}{\delta_4}) & \cdots & -\beta_2\alpha_3\alpha_4\cdots\alpha_{n-1}(\frac{\gamma_n}{\delta_n}) \\ 0 & \frac{1}{\delta_2} & -\frac{\theta_3}{\delta_3} & -\beta_3(\frac{\gamma_4}{\delta_4}) & \ddots & -\beta_3\alpha_4\cdots\alpha_{n-1}(\frac{\gamma_n}{\delta_n}) \\ 0 & 0 & \frac{1}{\delta_3} & -\frac{\theta_4}{\delta_4} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & -\beta_{n-1}(\frac{\gamma_n}{\delta_n}) \\ 0 & \cdots & 0 & 0 & \frac{1}{\delta_{n-1}} & -\frac{\theta_n}{\delta_n} \end{bmatrix}. \tag{1.13}$$

Again, a straightforward inspection of the corresponding A_{12} indicates that polynomials $\{r_k(x)\}$ of (1.12) form a H-1-q.s. system.

To sum up, the class of H-m-q.s. polynomials includes not only the well-studied classes of real orthogonal polynomials and the Szegö polynomials, but also several other interesting polynomial classes described in Sections 1.5.1 - 1.5.4. Hence, it is of interest to generalize the Björck-Pereyra algorithm to a polynomial-Vandermonde matrix $V_R(x)$ corresponding to a system of H-m-q.s. polynomials R. This is exactly what is done in the rest of the paper.

[5]Here the complex numbers $|\rho_k| \le 1$ are referred to as called *reflection coefficients*, and $\mu_k := \sqrt{1-|\rho_k|^2}$ if $|\rho_k| < 1$ and $\mu_k := 1$ if $|\rho_k| = 1$ are called *complementary parameters*.

2. The classical Björck-Pereyra algorithm. We begin by recalling the classical Björck-Pereyra algorithm. In [BP70], the authors derive a representation for the inverse $V(x)^{-1}$ of an $n \times n$ Vandermonde matrix as the product of bidiagonal matrices, that is,

$$V(x)^{-1} = U_1 \cdots U_{n-1} \cdot \tilde{L}_{n-1} \cdots \tilde{L}_1 \tag{2.1}$$

and used this result to solve the linear system $V(x)a = f$ by computing the solution vector

$$a = U_1 \cdots U_{n-1} \cdot \tilde{L}_{n-1} \cdots \tilde{L}_1 f$$

which solves the linear system in $\frac{5}{2}n^2$ operations. This is an order of magnitude improvement over Gaussian elimination, which is well known to require $\mathcal{O}(n^3)$ operations in general. This favorable complexity results from the fact that the matrices U_k and L_k are sparse. More specifically, the factors U_k and L_k are given by

$$U_k = \left[\begin{array}{c|cccc} I_{k-1} & & & & \\ \hline & 1 & -x_k & & \\ & & 1 & \ddots & \\ & & & \ddots & -x_k \\ & & & & 1 \end{array}\right], \tag{2.2}$$

$$\tilde{L}_k = \left[\begin{array}{c|ccc} I_k & & & \\ \hline & \frac{1}{x_{k+1}-x_1} & & \\ & & \ddots & \\ & & & \frac{1}{x_n - x_{n-k}} \end{array}\right] \cdot \left[\begin{array}{c|cccc} I_{k-1} & & & & \\ \hline & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{array}\right]. \tag{2.3}$$

In the next section we will present the new Björck-Pereyra-like algorithm for the most general case considered in this paper: the general Hessenberg case.

3. New Björck-Pereyra-like algorithm. General Hessenberg case. In this section we consider the linear system $V_R(x)a = f$, where $V_R(x)$ is the polynomial-Vandermonde matrix corresponding to the polynomial system R and the n distinct nodes x. No restrictions are placed on the polynomial system R at this point other than $\deg(r_k) = k$. The new algorithm is based on a decomposition of the inverse $V_R(x)^{-1}$ enabled by the following lemma. Herein we use the MATLAB convention $x_{i:j} = \left[\begin{array}{ccccc} x_i & x_{i+1} & \dots & x_{j-1} & x_j \end{array} \right]^T$ to denote a portion of the larger vector $x = \left[\begin{array}{ccccc} x_1 & x_2 & \dots & x_{n-1} & x_n \end{array} \right]^T$.

LEMMA 3.1. *Let $R = \{r_0(x), \dots, r_n(x)\}$ be an arbitrary system of polynomials as in (1.2), and denote $R_1 = \{r_0(x), \dots, r_{n-1}(x)\}$. Further let $x_{1:n} = (x_1, ..., x_n)$ be n distinct points. Then the inverse of $V_R(x_{1:n})$ admits a decomposition*

$$V_R(x_{1:n})^{-1} = U_1 \cdot \left[\begin{array}{cc} 1 & 0 \\ 0 & V_R(x_{2:n})^{-1} \end{array} \right] L_1, \tag{3.1}$$

with

$$U_1 = \left[\begin{array}{c|ccc} \frac{1}{\alpha_0} & & & \\ 0 & & & \\ & & C_{R_1}(r_{n-1}) - x_1 I & \\ \vdots & & & \\ \hline 0 & \cdots & 0 & \frac{1}{\alpha_{n-1}} \end{array}\right], \tag{3.2}$$

$$L_1 = \left[\begin{array}{cccc} 1 & & & \\ & \frac{1}{x_2 - x_1} & & \\ & & \ddots & \\ & & & \frac{1}{x_n - x_1} \end{array} \right] \left[\begin{array}{cccc} 1 & & & \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{array} \right]. \tag{3.3}$$

The proof of Lemma 3.1 is given in Section 6, but first we present its use in solving the linear system $V_R(x)a = f$.

3.1. Solving polynomial Vandermonde systems. Like the classical Björk-Pereyra algorithm, the recursive nature of (3.1) allows a decomposition of $V_R(x)^{-1}$ into $2n-2$ factors,

$$V_R(x)^{-1} = U_1 \cdot \left[\begin{array}{c|c} I_1 & \\ \hline & U_2 \end{array}\right] \cdots \left[\begin{array}{c|c} I_{n-2} & \\ \hline & U_{n-1} \end{array}\right] \cdot \left[\begin{array}{c|c} I_{n-2} & \\ \hline & L_{n-1} \end{array}\right] \cdots \left[\begin{array}{c|c} I_1 & \\ \hline & L_2 \end{array}\right] \cdot L_1, \tag{3.4}$$

with the lower and upper triangular factors given in (3.2), (3.3). The associated linear system can be solved by multiplying (3.4) by the right-hand side vector f.

It is emphasized that this decomposition is valid for any polynomial system R, however no computational savings are guaranteed. In order to have the desired computational savings, each multiplication of a matrix from (3.4) by a vector must be performed quickly.

The factors L_k are sparse as in the classical Björck-Pereyra algorithm, and thus multiplication by them is fast. However, unlike the classical Björck-Pereyra algorithm, the factors U_k are not sparse in general. In order to have a fast $\mathcal{O}(n^2)$ algorithm for solving the system $V_R(x)a = f$, it is necessary to be able to multiply each matrix in (3.4) by f in $\mathcal{O}(n)$ operations.

3.2. Differences between L_k and $\tilde{L}_k$. It is easy to see that the classical Vandermonde matrix V involving monomials and the polynomial Vandermonde matrix V_R are related via

$$V_R = V \cdot U,$$

where U is a change of basis matrix:

$$\begin{bmatrix} 1 & x & \cdots & x^{n-1} \end{bmatrix} \cdot U = \begin{bmatrix} r_0(x) & r_1(x) & \cdots & r_{n-1}(x) \end{bmatrix}.$$

Since U is upper triangular and since the LU factorization is unique, it follows that both matrices V and V_R share the same L factor that can be equivalently factored in different ways, e.g., using elementary factors (2.3) as in (2.1) or using elementary factors (3.3) as in (3.4).

Our choice of (3.3) is motivated by the heuristics that the latter uses the numbers $\{(x_2-x_1)^{-1}, \ldots, (x_n-x_1)^{-1}\}$. Since the same x_1 is used in each factor, (3.3) can potentially provide better accuracy if the so-called Leja ordering (that maximizes the separation between x_1 to the other nodes) is used ([RO91], [H90], [O03]). See Section 7 for more details.

4. Known special cases where the Björck-Pereyra-like algorithm is fast. We next present a detailed reduction of the algorithm presented in the previous section in several important special cases that were studied earlier by different authors.

4.1. Monomials. The classical Björck-Pereyra algorithm. Suppose the system of polynomials in the polynomial-Vandermonde matrix is simply a set of monomials; that is, $R = \{1, x, \ldots, x^{n-1}, x^n\}$. Then (1.2) becomes simply

$$r_0(x) = 1, \quad r_k(x) = x r_{k-1}(x), \quad k = 1, \ldots, n \tag{4.1}$$

and the corresponding confederate matrix is

$$C_R(r_n) = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}; \tag{4.2}$$

that is, $\alpha_k = 1$ for $k = 0, \ldots, n-1$. Inserting this and (4.2) into (3.2) yields (2.2), implying the factors U_k reduce to those of the classical Björck-Pereyra algorithm in this case. That is, the Vandermonde linear system $V_R(x)a = f$ can be solved via the factorization of (3.4):

$$a = V_R(x)^{-1} f = U_1 \cdot \left[\begin{array}{c|c} I_1 & \\ \hline & U_2 \end{array}\right] \cdots \left[\begin{array}{c|c} I_{n-2} & \\ \hline & U_{n-1} \end{array}\right] \cdot \left[\begin{array}{c|c} I_{n-2} & \\ \hline & L_{n-1} \end{array}\right] \cdots \left[\begin{array}{c|c} I_1 & \\ \hline & L_2 \end{array}\right] \cdot L_1 \cdot f. \tag{4.3}$$

Thus when the quasiseparable polynomials in question reduce to the monomials, the algorithm described reduces to the classical Björck-Pereyra algorithm. Due to the sparseness of the matrices involved, the overall cost of the algorithm is only $\frac{5}{2}n^2$.

4.2. Real orthogonal polynomials. The Higham algorithm. If the polynomial system under consideration satisfies the three-term recurrence relations

$$r_k(x) = (\alpha_k x - \delta_k) r_{k-1}(x) - \gamma_k r_{k-2}(x), \tag{4.4}$$

then the resulting confederate matrix is tridiagonal

$$C_R(r_n) = \begin{bmatrix} \frac{\delta_1}{\alpha_1} & \frac{\gamma_2}{\alpha_2} & 0 & \cdots & 0 \\ \frac{1}{\alpha_1} & \frac{\delta_2}{\alpha_2} & \ddots & \ddots & \vdots \\ 0 & \frac{1}{\alpha_2} & \ddots & \frac{\gamma_{n-1}}{\alpha_{n-1}} & 0 \\ \vdots & & \ddots & \frac{\delta_{n-1}}{\alpha_{n-1}} & \frac{\gamma_n}{\alpha_n} \\ 0 & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{\delta_n}{\alpha_n} \end{bmatrix} \tag{4.5}$$

which leads to the matrices U_k of the form

$$U_k = \left[\begin{array}{c|cccccc} \frac{1}{\alpha_0} & \frac{\delta_1}{\alpha_1} - x_k & \frac{\gamma_2}{\alpha_2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\alpha_1} & \frac{\delta_2}{\alpha_2} - x_k & \ddots & \ddots & \vdots \\ & 0 & \frac{1}{\alpha_2} & \ddots & \frac{\gamma_{k-1}}{\alpha_{k-1}} & 0 \\ \vdots & \vdots & & \ddots & \frac{\delta_{k-1}}{\alpha_{k-1}} - x_k & \frac{\gamma_k}{\alpha_k} \\ & 0 & \cdots & 0 & \frac{1}{\alpha_{k-1}} & \frac{\delta_k}{\alpha_k} - x_k \\ \hline 0 & & \cdots & & 0 & \frac{1}{\alpha_{n-j}} \end{array} \right]$$

Again, the factorization (4.3) uses the above to solve the linear system. The sparseness of these matrices allows computational savings, and the overall cost of the algorithm is again $\mathcal{O}(n^2)$.

In this case, the entire algorithm presented reduces to Algorithm 2.1 in [H90]. In particular, the multiplication of a vector by the matrices specified in (4.3) involving L_k and U_k can be seen as Stage I and Stage II in that algorithm, respectively.

4.3. Szegö polynomials. The [BEGKO07] algorithm. If the system of quasiseparable polynomials are the Szegö polynomials $\Phi^{\#} = \{\phi_0^{\#}(x), \dots, \phi_n^{\#}(x)\}$ represented by the reflection coefficients ρ_k and complimentary parameters μ_k (see [BC92]), then they satisfy the recurrence relations

$$\phi_0^{\#}(x) = 1, \quad \phi_1^{\#}(x) = \frac{1}{\mu_1} \cdot x\phi_0^{\#}(x) - \frac{\rho_1}{\mu_1}\phi_0^{\#}(x)$$

$$\phi_k^{\#}(x) = \left[\frac{1}{\mu_k} \cdot x + \frac{\rho_k}{\rho_{k-1}}\frac{1}{\mu_k}\right]\phi_{k-1}^{\#}(x) - \frac{\rho_k}{\rho_{k-1}}\frac{\mu_{k-1}}{\mu_k} \cdot x \cdot \phi_{k-2}^{\#}(x) \tag{4.6}$$

which are known to be associated to the almost unitary Hessenberg matrix

$$C_{\Phi^{\#}}(\phi_n^{\#}) = \begin{bmatrix} -\rho_1\rho_0^* & \cdots & -\rho_{n-1}\mu_{n-2}\dots\mu_1\rho_0^* & -\rho_n\mu_{n-1}\dots\mu_1\rho_0^* \\ \mu_1 & \ddots & -\rho_{n-1}\mu_{n-2}\dots\mu_2\rho_1^* & -\rho_n\mu_{n-1}\dots\mu_2\rho_1^* \\ 0 & \ddots & \vdots & \vdots \\ \vdots & & -\rho_{n-1}\rho_{n-2}^* & -\rho_n\mu_{n-1}\rho_{n-2}^* \\ 0 & \cdots & \mu_{n-1} & -\rho_n\rho_{n-1}^* \end{bmatrix} \tag{4.7}$$

In particular, if (4.7) is inserted into the factors (3.2) in (4.3), then the result is exactly that derived in [BEGKO07, (3.10) and (3.15)], where the nice properties of the matrix $C_{\Phi^{\#}}(\phi_n^{\#})$ were used to provide a computational speedup. Specifically, the algorithm is made fast by the factorization

$$C_{\Phi^{\#}}(\phi_n^{\#}) = G(\rho_1) \cdot G(\rho_2) \cdot \dots \cdot G(\rho_{n-1}) \cdot \tilde{G}(\rho_n), \tag{4.8}$$

where

$$G(\rho_j) = \mathrm{diag}\{I_{j-1}, \begin{bmatrix} \rho_j & \mu_j \\ \mu_j & -\rho_j^* \end{bmatrix}, I_{n-j-1}\}, \quad j = 1, 2, \ldots, n-1$$

and

$$\tilde{G}(\rho_n) = \mathrm{diag}\{I_{n-1}, \rho_n\}$$

see, for instance, [G82], [BC92], or [R95]. This gives an overall computational cost of $\mathcal{O}(n^2)$.

In the next section, we present a new special case which contains all previous special cases.

5. A new special case. Hessenberg-m-quasiseparable polynomials.

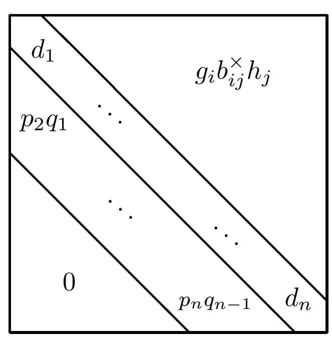
5.1. Rank definition. As stated in the introduction, a matrix A is called upper quasiseparable of order m if $\max(\mathrm{rank} A_{12}) = m$ where the maximum is taken over all symmetric partitions of the form

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

Also, a matrix $A = [a_{ij}]$ is called *upper Hessenberg* if all entries below the first subdiagonal are zeros; that is, $a_{ij} = 0$ if $i > j+1$. For brevity, we shall refer to order m upper quasiseparable Hessenberg matrices as *H-m-q.s. matrices*. Polynomials corresponding to H-m-q.s. matrices are called *H-m-q.s. polynomials*, or sometimes simply H-q.s. polynomials.

5.2. Generator definition. We next present an equivalent definition of an H-m-q.s. matrix in terms of its *generators*. The equivalence of these two definitions is well-known, see, e.g., in [EG991]. An $n \times n$ matrix $C_R(r_n)$ is called H-m-q.s. if it is of the form

$C_R(r_n) =$



(5.1)

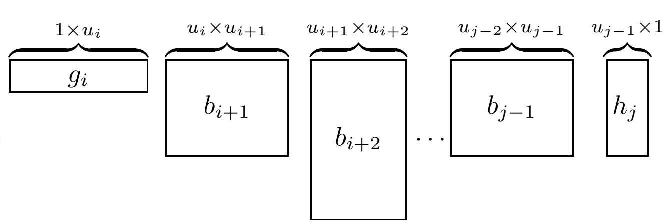
with

$$b_{ij}^{\times} = (b_{i+1}) \cdots (b_{j-1}), \quad b_{i,i+1} = I. \tag{5.2}$$

Here p_k, q_k, d_k are scalars, the elements g_k are row vectors of maximal size m, h_k are column vectors of maximal size m, and b_k are matrices of maximal size $m \times m$ such that all products make sense. The elements $\{p_k, q_k, d_k, g_k, b_k, h_k$ are called the *generators* of the matrix $C_R(r_n)$.

The elements in the upper part of the matrix $g_i b_{ij}^{\times} h_j$ are products of a row vector, a (possibly empty) sequence of matrices possibly of different sizes, and finally a column vector, as depicted here:

(5.3) $g_i b_{ij}^{\times} h_j =$



with $u_k \leq m$ for each $k = 1, \ldots, n-1$.

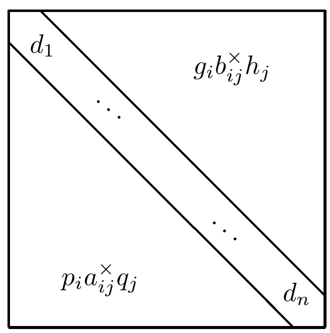
5.3. Fast multiplication using the quasiseparable structure. Here we show that in the special case of H-m-q.s. polynomials our Björck-Pereyra-like algorithms is fast, requiring $\mathcal{O}(n^2)$ operations. In view of (4.3) it suffices to have an algorithm for $\mathcal{O}(n)$ multiplication of a quasiseparable matrix by a vector since each matrix U_k contains a quasiseparable matrix as in (3.2). With such an algorithm, each multiplication in (3.4) could be implemented in $\mathcal{O}(n)$ operations, hence the total cost of computing the solution a would be $\mathcal{O}(n^2)$. The fast multiplication algorithm presented next is valid for a slightly more general class of quasiseparable matrices; specifically, the Hessenberg structure emphasized above is not essential here. To describe a more general result, we first give a slightly more general definition.

A matrix A is called (n_L, n_U)-quasiseparable if $\max(\text{rank} A_{12}) = n_U$ and $\max(\text{rank} A_{21}) = n_L$, where the maximum is taken over all symmetric partitions of the form

$$A = \left[\begin{array}{c|c} * & A_{12} \\ \hline A_{21} & * \end{array}\right] \tag{5.4}$$

Similar to the generator representation for upper quasiseparable matrices given above, arbitrary order quasiseparable matrices can be expressed in terms of generators as

$C_R(r_n) =$



with $a_{ij}^{\times} = (a_{i-1}) \cdots (a_{j+1})$, $a_{i+1,i} = I$, b_{ij} is as defined in (5.2). Now d_k are scalars, the elements p_k, g_k are row vectors of maximal size n_L and n_U respectively, q_k, h_k are column vectors of maximal size n_L and n_U respectively, and a_k, b_k are matrices of maximal size $n_L \times n_L$ and $n_U \times n_U$ respectively, such that all products make sense. The entries in the lower triangular part are defined by the product $p_i a_{ij}^{\times} q_j$, which has a similar form to that shown in (5.3).

Such a fast algorithm for multiplying a quasiseparable matrix by a vector is suggested by the following decomposition, valid for any (n_L, n_U)-quasiseparable matrix.

Proposition 5.1. *Let $C_R(r_n)$ be an $n \times n$ (n_L, n_U)-quasiseparable matrix specified by its generators as in Section 5.2. Then $C_R(r_n)$ admits the decomposition*

$$C_R(r_n) = L + D + U$$

where

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & p_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & p_n \end{bmatrix} \left[\begin{array}{ccc|c} 0 & \cdots & 0 & 0 \\ \hline & & & 0 \\ & \widetilde{A}^{-1} & & \vdots \\ & & & 0 \end{array}\right] \begin{bmatrix} q_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & q_{n-1} & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_n \end{bmatrix}$$

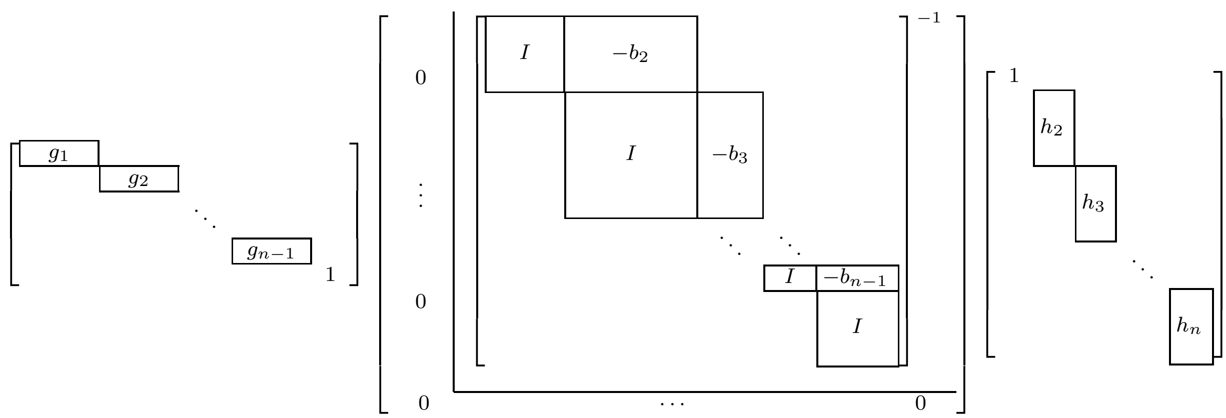
$$
(5.5) \qquad U = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & g_{n-1} & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \left[\begin{array}{c|ccc} 0 & & & \\ \vdots & & \widetilde{B}^{-1} & \\ 0 & & & \\ \hline 0 & 0 & \cdots & 0 \end{array}\right] \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & h_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & h_n \end{bmatrix}
$$

with

$$
\widetilde{A} = \begin{bmatrix} I & & & \\ -a_2 & \ddots & & \\ & \ddots & I & \\ & & -a_{n-1} & I \end{bmatrix}, \qquad \widetilde{B} = \begin{bmatrix} I & -b_2 & & \\ & \ddots & \ddots & \\ & & I & -b_{n-1} \\ & & & I \end{bmatrix}
$$

and I represents the identity matrix of appropriate size.

We emphasize that the diagonal matrices in Proposition 5.1 are block diagonal matrices, although in the 1-q.s. case they are actually diagonal. To illustrate this, the product forming the matrix U has the following form: $U =$



In light of this decomposition, we see that the matrix-vector product is reduced to five diagonal scalings and two back-substitutions with bidiagonal matrices. The justification of the proposition follows from the next simple lemma, which can be seen by direct confirmation.

LEMMA 5.2. *Let $b_k, k = 1, \ldots, m-1$ be matrices of sizes $u_k \times u_{k+1}$ (with u_1, u_m arbitrary) then*

$$
\begin{bmatrix} I_{n_1} & -b_1 & & & \\ & I_{n_2} & -b_2 & & \\ & & \ddots & \ddots & \\ & & & I_{n_{m-1}} & -b_{m-1} \\ & & & & I_{n_m} \end{bmatrix}^{-1} = \begin{bmatrix} I_{n_1} & b_1 & b_1b_2 & b_1b_2b_3 & \cdots & b_1b_2b_3\cdots b_{m-1} \\ & I_{n_2} & b_2 & b_2b_3 & \cdots & b_2b_3\cdots b_{m-1} \\ & & \ddots & \ddots & & \vdots \\ & & & I_{n_{m-2}} & b_{m-2} & b_{m-2}b_{m-1} \\ & & & & I_{n_{m-1}} & b_{m-1} \\ & & & & & I_{n_m} \end{bmatrix}
$$

We next consider the computational cost of the multiplication algorithm suggested by this proposition. Let $C_R(r_n)$ be a quasiseparable matrix of order (n_L, m) whose generators are of the following sizes:

Generator	p_k	a_k	q_k	d_k	g_k	b_k	h_k
Size	$1 \times l_{k-1}$	$l_k \times l_{k-1}$	$l_k \times 1$	1×1	$1 \times u_k$	$u_{k-1} \times u_k$	$u_{k-1} \times 1$

for numbers $l_k, u_k, k = 1, \ldots, n$. Define also $l_0 = u_0 = 0$. Then it can be seen that the computational cost is

$$
c = 3n + 2\sum_{k=1}^{n-1}(u_k + l_k + u_k u_{k-1} + l_k l_{k-1})
$$

flops (additions plus multiplications). Since $l_k \le n_L$ and $u_k \le n_U$ for $k = 1, \dots, n$, we also have

$$c \le 3n + 2(n-1)(n_U + n_L + n_U^2 + n_L^2).$$

Thus, for values of n_L and n_U much less than n, the quasiseparability is useful as the multiplication can be carried out in $\mathcal{O}(n)$ arithmetic operations.

Note additionally that the implementation of the algorithm suggested by the above decomposition coincides with the algorithm derived differently in [EG992] for the same purpose.

5.4. Special choices of generators. The new Björck-Pereyra algorithm is based on the decomposition (3.4) and the fast matrix-vector product algorithm of Section 5.3. The input of the latter algorithm is a generator of the corresponding H-m-q.s. matrix (5.1). However, in many examples, the generators of (5.1) are not given explicitly, but rather by the recurrence relations of the corresponding H-m-q.s. polynomials.

We next provide a detailed conversion between two different representations, i.e, Table 5.1 lists formulas to compute the generators from the recurrence relations coefficients for all special cases considered above.

Polynomials	p_k	q_k	d_k	g_k	b_k	h_k
Monomials (4.1)	1	1	0	0	0	1
Real orth. (4.4)	1	$1/\alpha_k$	δ_k/α_k	γ_k/α_k	0	1
Szegö (4.6)	1	μ_k	$-\rho_k\rho_{k-1}^*$	ρ_{k-1}^*	μ_{k-1}	$-\mu_{k-1}\rho_k$
Gen. 3-term (1.8)	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 (1.10)	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 (1.12)	1	$1/\delta_k$	$-\frac{\theta_k}{\delta_k}$	β_k	α_k	$-\frac{\gamma_k}{\delta_k}$

Table 5.1. Specific choices of generators resulting in various special cases.

To illustrate a set of generators for a higher order matrix, the next example considers the m-recurrent polynomials of Section 1.5.1.

Example 5.3. *Consider the system of polynomials $\{r_0(x), \dots, r_5(x)\}$ that are 4-recurrent; that is, they satisfy (1.6) with $m = 4$:*

$$r_k(x) = (\alpha_k x - a_{k-1,k}) \cdot r_{k-1}(x) - a_{k-2,k} \cdot r_{k-2}(x) - a_{k-3,k} \cdot r_{k-3}(x)$$

The corresponding confederate matrix is the following 5×5 matrix of the form in (1.7) with $m = 3$:

$$A = \begin{bmatrix} \frac{a_{0,1}}{\alpha_1} & \frac{a_{0,2}}{\alpha_2} & \frac{a_{0,3}}{\alpha_3} & 0 & 0 \\ \frac{1}{\alpha_1} & \frac{a_{1,2}}{\alpha_2} & \frac{a_{1,3}}{\alpha_3} & \frac{a_{1,4}}{\alpha_4} & 0 \\ 0 & \frac{1}{\alpha_2} & \frac{a_{2,3}}{\alpha_3} & \frac{a_{2,4}}{\alpha_4} & \frac{a_{2,5}}{\alpha_5} \\ 0 & 0 & \frac{1}{\alpha_3} & \frac{a_{3,4}}{\alpha_4} & \frac{a_{3,5}}{\alpha_5} \\ 0 & 0 & 0 & \frac{1}{\alpha_4} & \frac{a_{4,5}}{\alpha_5} \end{bmatrix}$$

It can readily be seen that the matrices given by

$$g_1 = \begin{bmatrix} a_{0,2} & a_{0,3} \end{bmatrix}, \quad g_2 = \begin{bmatrix} a_{1,4} & a_{1,3} \end{bmatrix}, \quad g_3 = \begin{bmatrix} a_{2,4} & a_{2,5} \end{bmatrix}, \quad g_4 = \begin{bmatrix} 0 & a_{3,5} \end{bmatrix}$$

$$b_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad b_3 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad b_4 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} \frac{1}{\alpha_2} \\ 0 \end{bmatrix}, \quad h_3 = \begin{bmatrix} 0 \\ \frac{1}{\alpha_3} \end{bmatrix}, \quad h_4 = \begin{bmatrix} \frac{1}{\alpha_4} \\ 0 \end{bmatrix}, \quad h_5 = \begin{bmatrix} 0 \\ \frac{1}{\alpha_5} \end{bmatrix},$$

are generators of the matrix A.

6. Derivation of the new Björck-Pereyra-like algorithm. In this section the algorithm presented is derived and the main enabling lemma is proved. The section begins with some background material.

6.1. Associated (generalized Horner) polynomials. Following [KO97] define the *associated polynomials* $\widehat{R} = \{\widehat{r}_0(x), \ldots, \widehat{r}_n(x)\}$ for a given system of polynomials $R = \{r_0(x), \ldots, r_n(x)\}$ via the relation

$$\frac{r_n(x) - r_n(y)}{x - y} = \begin{bmatrix} r_0(x) & r_1(x) & r_2(x) & \cdots & r_{n-1}(x) \end{bmatrix} \cdot \begin{bmatrix} \hat{r}_{n-1}(y) \\ \hat{r}_{n-2}(y) \\ \vdots \\ \hat{r}_1(y) \\ \hat{r}_0(y) \end{bmatrix}, \tag{6.1}$$

with additionally $\widehat{r}_n(x) = r_n(x)$.

However, before proceeding we first clarify the existence of such polynomials. Firstly, the polynomials associated with the monomials exist. Indeed, if P is the system of $n+1$ polynomials $P = \{1, x, x^2, ..., x^{n-1}, r_n(x)\}$, then

$$\frac{r_n(x) - r_n(y)}{x - y} = \begin{bmatrix} 1 & x & x^2 & \cdots & x^{n-1} \end{bmatrix} \cdot \begin{bmatrix} \hat{p}_{n-1}(y) \\ \hat{p}_{n-2}(y) \\ \vdots \\ \hat{p}_1(y) \\ \hat{p}_0(y) \end{bmatrix} = \sum_{i=0}^{n-1} x^i \cdot \hat{p}_{n-1-i}(y), \tag{6.2}$$

and in this case the associated polynomials $\widehat{P}$ can be seen to be the classical Horner polynomials (see, e.g., [KO97, Section 3.]).

Secondly, given a system of polynomials $R = \{r_0(x), r_1(x), \ldots, r_{n-1}(x), r_n(x)\}$, there is a corresponding system of polynomials $\hat{R} = \{\hat{r}_0(x), \hat{r}_1(x), \ldots, \hat{r}_{n-1}(x), \hat{r}_n(x)\}$ (with $\hat{r}_n(x) = r_n(x)$) satisfying (6.1). One can see that, given a polynomial system R with $\deg(r_k) = k$, the polynomials in R can be obtained from the monomial basis by

$$\begin{bmatrix} 1 & x & x^2 & \cdots & x_{n-1} \end{bmatrix} S = \begin{bmatrix} r_0(x) & r_1(x) & r_2(x) & \cdots & r_{n-1}(x) \end{bmatrix} \tag{6.3}$$

where S is an $n \times n$ upper triangular invertible matrix capturing the recurrence relations of the polynomial system R. Inserting SS^{-1} into (6.2) between the row and column vectors and using (6.3), we see that the polynomials associated with R are

$$\begin{bmatrix} \hat{r}_{n-1}(y) \\ \hat{r}_{n-2}(y) \\ \vdots \\ \hat{r}_1(y) \\ \hat{r}_0(y) \end{bmatrix} = S^{-1} \begin{bmatrix} \hat{p}_{n-1}(y) \\ \hat{p}_{n-2}(y) \\ \vdots \\ \hat{p}_1(y) \\ \hat{p}_0(y) \end{bmatrix} \tag{6.4}$$

where $\widehat{P} = \{\widehat{p}_0(x), \ldots, \widehat{p}_{n-1}(x)\}$ are the classical Horner polynomials and S is from (6.3).

The following lemma will be needed in the proof presented below.

LEMMA 6.1. *Let $R = \{r_0(x), \ldots, r_{n-1}(x)\}$ be a system of polynomials satisfying (1.2), and for $k = 1, 2, \ldots, n-1$ denote by $R^{(k)}$ the system of polynomials $R^{(k)} = \{\hat{r}_0^{(k)}(x), \ldots, \hat{r}_k^{(k)}(x)\}$ associated with the truncated system $\{r_0(x), \ldots, r_k(x)\}$. Then*

$$\begin{bmatrix} \hat{r}_0^{(1)}(x) & \hat{r}_1^{(2)}(x) & \cdots & \hat{r}_{n-2}^{(n-1)}(x) \\ & \hat{r}_0^{(2)}(x) & \cdots & \hat{r}_{n-3}^{(n-1)}(x) \\ & & \ddots & \vdots \\ & & & \hat{r}_0^{(n-1)}(x) \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{\alpha_1} & -x + \frac{a_{1,2}}{\alpha_2} & \cdots & \frac{1}{\alpha_{n-1}} a_{1,n-1} \\ & \frac{1}{\alpha_2} & \cdots & \frac{1}{\alpha_{n-1}} a_{2,n-1} \\ & & \ddots & \vdots \\ & & & -x + \frac{a_{n-2,n-1}}{\alpha_{n-1}} \\ & & & \frac{1}{\alpha_{n-1}} \end{bmatrix} \tag{6.5}$$

Proof. From [KO97] we have the formula

$$C_{\hat{R}}(\hat{r}_n) = \tilde{I} \cdot C_R(r_n)^T \cdot \tilde{I}, \qquad (\text{with} \quad \hat{r}_n(x) = r_n(x)), \tag{6.6}$$

where $\tilde{I}$ is the antidiagonal matrix, which provides a relation between the confederate matrix of a polynomial system R and that of the polynomials associated with R. From this we have the following n-term recurrence relations for the truncated associated polynomials:

$$\hat{r}_m^{(k)}(x) = \alpha_m \left[\left(x - \frac{a_{m,m+1}}{\alpha_{m+1}} \right) \hat{r}_{m-1}^{(k)} - \frac{a_{m,m+2}}{\alpha_{m+2}} \hat{r}_{m-2}^{(k)} - \cdots - \frac{a_{m,k}}{\alpha_k} \hat{r}_0^{(k)} \right], \quad m = 1, \ldots, k-1, \tag{6.7}$$

with

$$\hat{r}_0^{(k)} = 1/\alpha_k. \tag{6.8}$$

Now consider the product

$$\begin{bmatrix} \frac{1}{\alpha_1} & -x + \frac{a_{1,2}}{\alpha_2} & \cdots & \cdots & \frac{a_{1,n-1}}{\alpha_{n-1}} \\ & \ddots & \ddots & & \vdots \\ & & \frac{1}{\alpha_i} & -x + \frac{a_{i,i+1}}{\alpha_{i+1}} \quad \cdots & \frac{a_{i,n-1}}{\alpha_{n-1}} \\ & & & \ddots & -x + \frac{a_{n-2,n-1}}{\alpha_{n-1}} \\ & & & & \frac{1}{\alpha_{n-1}} \end{bmatrix} \begin{bmatrix} \widehat{r}_0^{(1)}(x) & \cdots & \widehat{r}_{j-1}^{(j)}(x) & \cdots & \widehat{r}_{n-2}^{(n-1)}(x) \\ & & \widehat{r}_{j-2}^{(j)}(x) & & \widehat{r}_{n-3}^{(n-1)}(x) \\ & \ddots & \vdots & & \vdots \\ & & \widehat{r}_0^{(j)}(x) & & \widehat{r}_1^{(n-1)}(x) \\ & & & \ddots & \widehat{r}_0^{(n-1)}(x) \end{bmatrix}. \tag{6.9}$$

The (i,j) entry of this product defined by the highlighted row and column can be seen as (6.7) with $k = j, m = j - i$ if $i \neq j$ and (6.8) with $k = i, m = 0$ if $i = j$. Thus this product is the identity, implying (6.5). ◻

With this completed, next is the proof of Lemma 3.1 from Section 3.

Proof. Performing one step of Gaussian elimination on $V_R(x_{1:n})$ yields

$$V_R(x_{1:n}) = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & & \ddots & \\ 1 & & & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & & \\ & x_2 - x_1 & & \\ & & \ddots & \\ & & & x_n - x_1 \end{bmatrix}.$$

$$\cdot \begin{bmatrix} 1 & 0 \\ 0 & \bar{R} \end{bmatrix} \cdot \left[\begin{array}{c|ccc} r_0(x_1) & r_1(x_1) & \cdots & r_{n-1}(x_1) \\ \hline 0 & & I & \end{array} \right], \tag{6.10}$$

where the matrix $\bar{R}$ has (i,m)-entry $\bar{R}_{i,m} = \frac{r_{m+1}(x_{i+1}) - r_1(x_1)}{x_{i+1} - x_1}$; that is, $\bar{R}$ consists of divided differences. By the discussion above, associated with the system R is the system $\widehat{R} = \{\hat{r}_0(x), \ldots, \hat{r}_n(x)\}$. Following the notation of Lemma 6.1, denote by $\widehat{R}^{(k)} = \{\hat{r}_0^{(k)}(x), \ldots, \hat{r}_k^{(k)}(x)\}$ the system of polynomials associated with the truncated system $\{r_0(x), ..., r_k(x)\}$. By the definition of the associated polynomials we have for $k = 1, 2, \ldots, n-1$

$$\frac{r_k(x) - r_k(y)}{x - y} = \begin{bmatrix} r_0(x) & r_1(x) & r_2(x) & \cdots & r_{k-1}(x) \end{bmatrix} \cdot \begin{bmatrix} \hat{r}_{k-1}^{(k)}(y) \\ \hat{r}_{n-2}^{(k)}(y) \\ \vdots \\ \hat{r}_1^{(k)}(y) \\ \hat{r}_0^{(k)}(y) \end{bmatrix} = \sum_{i=0}^{k-1} r_i(x) \cdot \hat{r}_{k-1-i}^{(k)}(y).$$

Finally, denoting by $\widehat{R}^{(k)} = \{\hat{r}_0^{(k)}(x), \ldots, \hat{r}_k^{(k)}(x)\}$ the system of polynomials associated with the truncated system $\{r_0(x), \ldots, r_k(x)\}$ we can further factor $\bar{R}$ as

$$\bar{R} = V_R(x_{1:n}) \cdot \begin{bmatrix} \hat{r}_0^{(1)}(x_j) & \hat{r}_1^{(2)}(x_j) & \cdots & \hat{r}_{n-2}^{(n-1)}(x_j) \\ & \hat{r}_0^{(2)}(x_j) & \cdots & \hat{r}_{n-3}^{(n-1)}(x_j) \\ & & \ddots & \vdots \\ & & & \hat{r}_0^{(n-1)}(x_j) \end{bmatrix}. \tag{6.11}$$

The last matrix on the right-hand side of (6.11) can be inverted by Lemma 6.1. Therefore, inverting (6.10) and substituting (6.5) results in (3.1) as desired. □

7. Numerical Illustrations. We report here several results of our preliminary numerical experiments to indicate that in the generic case the behavior of the generalized algorithms is consistent with the conclusions reported earlier for the known Björck-Pereyra-like algorithms.

The algorithm has been implemented in MATLAB version 7, which uses double precision. These results were compared with exact solutions calculated using the MATLAB Symbolic Toolbox command `vpa()`, which allows software-implemented precision of arbitrary numbers of digits. The number of digits was set to 64, however in cases where the condition number of the coefficient matrix exceeded 10^{30}, this was raised to 100 digits to maintain accuracy.

It is known (see [RO91], [H90]) that reordering the nodes for polynomial Vandermonde matrices, which corresponds to a permutation of the rows, can affect the accuracy of related algorithms. In particular, ordering the nodes according to the *Leja ordering*

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

(see [RO91], [H90], [O03]) improves the performance of many similar algorithms. We include experiments with and without the use of Leja ordering (if the Leja ordering is not used, the nodes are ordered randomly). A counterpart of this ordering is known for Cauchy matrices, see [BKO02].

In all experiments, we compare the forward accuracy of the algorithm, defined by

$$e = \frac{\|x - \hat{x}\|_2}{\|x\|_2}$$

where $\hat{x}$ is the solution computed by each algorithm in MATLAB in double precision, and x is the exact solution. In the tables, BP-QS denotes the proposed Björck-Pereyra like algorithm with a random ordering of the nodes, and BP-QS-L denotes the same algorithm using the Leja ordering. The factors L_k from (3.3) were used. GE indicates MATLAB's Gaussian elimination. Finally, cond(V) denotes the condition number of the matrix V computed via the MATLAB command `cond()`.

Experiment 1. In Table 7.1, the values for the generators were chosen randomly on $(-1, 1)$, similarly for the entries of the right hand side vector. The nodes x_k were selected equidistant on $(-1, 1)$ via the formula

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

We test the accuracy of the algorithm for various sizes n of matrices generated in this way.

n	`cond(V)`	GEPP	BP-QS	BP-QS-L
10	6.9e+06	8.7e-15	1.9e-14	1.6e-15
	3.5e+08	1.9e-14	5.3e-15	8.9e-16
	1.9e+10	7.1e-15	6.0e-16	6.4e-16
15	1.5e+10	4.4e-12	3.5e-12	6.7e-15
	7.7e+13	5.8e-12	1.4e-13	1.3e-15
	5.9e+15	3.1e-11	4.3e-13	5.7e-16
20	6.0e+17	1.0e-09	1.4e-11	4.6e-15
	2.2e+18	9.6e-14	8.5e-12	1.2e-15
	1.6e+22	6.2e-11	1.1e-11	2.3e-15
25	1.6e+20	8.0e-08	4.3e-11	4.4e-16
	1.0e+22	1.3e-08	1.1e-10	1.3e-15
	1.0e+26	8.8e-07	1.5e-10	3.2e-15
30	9.1e+18	1.2e-06	4.3e-06	1.2e-14
	8.0e+23	5.0e-08	3.3e-09	1.5e-15
	1.9e+24	5.8e-02	5.6e-10	4.4e-15
35	9.8e+23	9.3e-01	1.2e-06	2.0e-15
	7.5e+28	1.6e-03	7.1e-08	1.7e-15
	1.8e+29	1.1e-02	4.2e-06	1.7e-15
40	2.6e+25	8.6e-02	1.1e-06	8.6e-15
	2.1e+29	2.9e-02	1.4e-06	4.8e-15
	1.0e+33	1.0e+00	2.2e-05	2.4e-16
45	4.5e+31	1.0e+00	8.4e-05	2.0e-15
	9.2e+36	1.2e+00	3.2e-05	3.0e-15
	5.9e+38	1.1e+00	2.2e-04	5.2e-16
50	3.3e+37	1.0e+00	6.9e-03	1.2e-13
	2.8e+41	4.0e-01	4.8e-03	2.3e-13
	8.7e+45	1.0e+00	1.6e-02	6.3e-14

Table 7.1. Equidistant nodes on $(-1, 1)$.

Notice that as the size of the matrices involved rises, so does the condition number of the matrices, and hence as expected, the performance of Gaussian elimination declines. The performance of the proposed algorithm with a random ordering of the nodes is an improvement over that of GE, however using the Leja ordering gives a dramatic improvement in performance in this case.

Experiment 2. In Table 7.2, the values for the generators and entries of the right hand side vector were chosen as in Experiment 1, and the nodes x_k were selected clustered on $(-1,1)$ via the formula

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

Again we test the accuracy for various $n \times n$ matrices generated in this way.

n	`cond(V)`	GEPP	BP-QS	BP-QS-L
10	8.6e+10	1.7e-12	9.0e-15	5.5e-16
	1.2e+11	1.9e-12	1.5e-14	1.0e-15
	3.3e+12	1.7e-13	1.6e-14	3.6e-16
15	1.3e+15	1.3e-11	3.0e-13	3.7e-15
	1.9e+16	5.1e-11	6.4e-13	3.4e-16
	1.5e+17	1.4e-11	8.6e-14	2.3e-15
20	9.7e+17	6.0e-08	1.5e-10	8.5e-14
	2.1e+18	5.4e-08	1.7e-12	8.9e-15
	1.1e+23	5.0e-08	3.1e-11	7.9e-16
25	1.8e+20	7.2e-04	6.9e-10	7.7e-14
	4.5e+20	4.8e-03	1.3e-09	5.9e-14
	4.0e+22	1.9e-03	2.9e-10	6.1e-15
30	9.1e+22	1.1e+00	6.1e-09	6.4e-15
	4.9e+24	1.0e+00	2.7e-10	2.2e-15
	1.0e+26	1.0e+00	2.9e-09	2.9e-12
35	2.1e+27	7.8e-01	2.5e-07	1.6e-14
	1.3e+28	1.0e+00	8.6e-09	2.9e-09
	4.7e+33	1.0e+00	8.3e-07	7.4e-11
40	7.5e+27	1.0e+00	2.1e-05	5.1e-08
	1.9e+33	1.0e+00	8.5e-06	2.5e-09
	3.3e+37	1.0e+00	3.8e-05	1.1e-12
45	1.8e+33	1.0e+00	6.2e-04	8.2e-04
	4.4e+34	1.0e+00	1.3e-03	1.0e-06
	1.2e+40	1.0e+00	2.0e-04	2.3e-08
50	6.5e+32	1.0e+00	7.3e-03	3.2e-06
	2.8e+36	1.0e+00	1.2e-03	5.6e-04
	1.5e+46	1.0e+00	3.1e-03	7.8e-04

Table 7.2. Clustered nodes on $(-1,1)$.

In this experiment, again the condition number rises with the size of the matrix, and as expected Gaussian elimination gives less accuracy as this increases. The proposed algorithm gives an improvement in this case as well, and the Leja ordering again gives an improvement.

Experiment 3. In [H87], [CF88], [BKO99] it was shown that the behavior of BKO-type algorithms can depend on the direction of the right hand side vector. We include a similar experiment here, and the outcome is consistent with observations in [CF88] and [BKO99]. This is illustrated in Table 7.3, which shows the results for a 30×30 matrix generated by a fixed set of generators and nodes on the unit disc, and the results of applying the various algorithms to solve the system with each (left) singular vector as the right hand side. In other words, for the singular value decomposition of V_R

$$V_R = U\Sigma V^T = \left[\begin{array}{c|c|c|c} u_1 & u_2 & \cdots & u_n \end{array} \right] \Sigma \left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array} \right]^T$$

we solve 30 systems of the form $V_R x = u_k$, where as depicted u_k is the k-th column of U.

singular vector	GEPP	BP-QS	BP-QS-L
u_1	4.7e-01	5.7e-02	2.7e+01
u_2	1.3e+00	2.8e-02	2.1e+00
u_3	1.6e+00	1.8e-02	1.9e+00
u_4	3.8e+00	3.6e-02	2.3e+00
u_5	5.4e+00	1.4e-02	1.6e+00
u_6	2.9e+00	6.3e-02	6.6e-02
u_7	5.3e+00	9.7e-03	2.0e-02
u_8	1.6e+00	2.1e-03	2.0e-03
u_9	5.5e-01	2.4e-04	6.6e-04
u_{10}	1.4e-01	1.4e-05	7.7e-05
u_{11}	8.1e-05	2.1e-09	3.6e-09
u_{12}	1.7e-05	5.8e-10	6.4e-10
u_{13}	7.9e-06	2.7e-10	9.4e-11
u_{14}	8.6e-06	2.7e-11	3.1e-11
u_{15}	5.8e-06	6.4e-11	6.7e-11
u_{16}	1.1e-05	1.8e-13	9.6e-13
u_{17}	1.1e-05	1.6e-15	9.6e-16
u_{18}	1.1e-05	2.5e-15	4.4e-15
u_{19}	1.1e-05	6.1e-15	2.1e-15
u_{20}	1.2e-05	8.5e-15	4.1e-14
u_{21}	1.3e-05	1.8e-15	4.4e-16
u_{22}	1.4e-05	4.7e-15	3.8e-15
u_{23}	1.3e-05	1.1e-14	1.5e-15
u_{24}	1.9e-05	2.2e-14	8.1e-15
u_{25}	5.2e-05	6.7e-14	5.0e-14
u_{26}	1.4e-05	1.9e-15	2.6e-15
u_{27}	1.3e-05	1.4e-15	2.2e-15
u_{28}	1.2e-05	2.9e-15	1.1e-15
u_{29}	1.2e-05	6.0e-15	2.5e-15
u_{30}	1.2e-05	7.3e-16	4.9e-16

Table 7.3. Dependence on the direction using left singular vectors.

The outcome is consistent with observations in [CF88] and [BKO99] for Björck-Pereyra-type algorithms for the classical Vandermonde and Cauchy matrices.

The above typical experiments describe generic H-q.s. cases, and they are preliminary. We conclude this section with two comments about further possible work. Firstly, experience indicates that numerical properties of general polynomial algorithms may be quite different even for two classical special cases: real orthogonal polynomials and Szegö polynomials. Hence it is of interest to study the numerical behavior of the new algorithm for different special cases, e.g., four examples described in Sections 1.5.1 - 1.5.4.

Secondly, there is an ongoing work [BOS] on studying the accuracy of fast algorithms for multiplying a quasiseparable matrix by a vector. There are several alternatives to the method described above in Section

5.3, and one of them is provably stable. When the work [BOS] is completed, it will be of interest to study the accuracy of our algorithm based on (3.4) combined with new methods for computing q.s. matrix-vector product.

However, already at this point the results of all of our experiments are fully consistent with the conclusions made in [BP70], [CF88], [H90], [RO91], [BKO99], [O03] and [BEGKO07] for similar algorithms. Specifically, there are examples, even in the most generic H-q.s. case, in which the Björck-Pereyra-type algorithms can yield a very high forward accuracy.

8. Conclusion. In this paper we generalized the well known Björck-Pereyra algorithm to polynomial-Vandermonde matrices. The efficiency of the algorithm depends on the structure of the corresponding confederate matrix (a Hessenberg matrix capturing the recurrence relations). In the case where the polynomial system is H-m-q.s. with small m (i.e., when the confederate matrix is quasiseparable) the algorithm has a favorably complexity of $\mathcal{O}(n^2)$ operations, which is an order of magnitude improvement over the standard methods. Initial numerical experiments indicate that in many cases the algorithm provides better forward accuracy than the one of Gaussian elimination. This observation is fully consistent with the experience of our colleagues reported in several previous papers.

REFERENCES

[BC92] 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.

[BEGOT06] T.Bella, Y.Eidelman, I.Gohberg, V.Olshevsky, and E.Tyrtyshnikov *Fast inversion of Hessenberg-quasiseparable Vandermonde matrices*, submitted.

[BEGKO07] 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*, to appear, Linear Algebra and Its Applications.

[BKO99] T.Boros, T.Kailath and V.Olshevsky, *Fast Bjorck-Pereyra-type algorithm for parallel solution of Cauchy linear equations*, Linear Algebra and Its Applications, 302-303 (1999), p.265-293.

[BKO02] T.Boros, T.Kailath, V.Olshevsky, *Pivoting and backward stability of fast algorithms for solving Cauchy linear equations*, Special issue on structured and infinite systems of linear equations. Linear Algebra Appl. 343/344 (2002), 63–99.

[BOS] T. Bella, V. Olshevsky, and M. Stewart, *Accuracy of several fast algorithms to multiply a quasiseparable matrix by a vector*, in preparation.

[BP70] A.Björck and V.Pereyra, *Solution of Vandermonde Systems of Equations*, Math. Comp., **24** (1970), 893-903.

[CF88] T.Chan and D.Foulser, *Effectively well-conditioned linear systems*, SIAM J. Sci. Stat. Computations, 9 (1988), 963 – 969.

[EG991] Y. Eidelman and I. Gohberg, *On a new class of structured matrices*, Integral Equations and Operator Theory, **34** (1999), 293-324.

[EG992] Y. Eidelman and I. Gohberg, *Linear complexity inversion algorithms for a class of structured matrices*, Integral Equations and Operator Theory, **35** (1999), 28-52.

[EG02] 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.

[EGO05] 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.

[G82] 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, Moskow University Press, 1982.
English translation in : J. Comput. and Appl. Math., **46**(1993), 183-198.

[G48] L.Y.Geronimus, *Polynomials orthogonal on a circle and their applications*, Amer. Math. Translations, **3** p.1-78, 1954 (Russian original 1948).

[GO94] I.Gohberg and V.Olshevsky, *Fast inversion of Chebyshev-Vandermonde matrices*, Numerische Mathematik, **67**, No. 1 (1994), 71 – 92.

[GS58] U.Grenader and G.Szegö, *Toeplitz forms and Applications*, University of California Press, 1958.

[GVL96] G.Golub and C.van Loan. Matrix Computations. Johns Hopkins University Press, 3rd edition, 1996.

[H87] N.J.Higham, *Error analysis of the Björck–Pereyra algorithms for solving Vandermonde systems*, Numerische mathematic, 50 (1987), 613 – 632.

[H90] N.J.Higham, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl., **11**(1) (1990), 23–41.

[KO97] T.Kailath and V.Olshevsky, *Displacement structure approach to polynomial Vandermonde and related matrices*, Linear Algebra and Its Applications, **261** (1997), 49-90.

[MB79] J.Maroulas and S.Barnett, Polynomials with respect to a general basis. I. Theory, J. of Math. Analysis and Appl., **72** (1979), 177 -194.

[O98] V.Olshevsky, *Eigenvector computation for almost unitary Hessenberg matrices and inversion of Szegö-Vandermonde matrices via discrete transmission lines,* Linear Algebra and Its Applications, 285 (1998), 37-67.

[O01] V.Olshevsky, *Associated polynomials, unitary Hessenberg matrices and fast generalized Parker-Traub and Bjorck-Pereyra algorithms for Szegö-Vandermonde matrices,* "Structured Matrices: Recent Developments in Theory and Computation," 67-78, (D.Bini, E. Tyrtyshnikov, P. Yalamov., Eds.), 2001, NOVA Science Publ.

[O03] 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.

[R90] L.Reichel, *Newton interpolation at Leja points*, BIT, 30(1990), 23 – 41.

[R95] P.A.Regalia, *Adaptive IIR filtering in signal processing and control*, Marcel Dekker, New York, 1995.

[RO91] L.Reichel and G.Opfer, *Chebyshev-Vandermonde systems*, Math. of Comp., **57** (1991), 703-721.

[T94] EE Tyrtyshnikov. *How bad are Hankel matrices?* Numerische Mathematik, (1994) 67:261–269

[TG81] W.Tang and G.Golub, *The block decomposition of a Vandermonde matrix and its aplications*, BIT, **21** (1981), 505-517.