Unitary Hessenberg matrices and the generalized Parker-Forney-Traub algorithm for inversion of Szegö-Vandermonde matrices*

V.Olshevsky†

Abstract

It is well-known that the *Horner polynomials* (sometimes called the associated polynomials) describe the structure of the inverses of Vandermonde matrices $V(x) = \left[\begin{array}{c} x_i^{j-1} \end{array} \right]$. This description led to the fast $O(n^2)$ inversion algorithm discovered independently by Parker, Forney, Traub and many others. In this paper we show how the *generalized Horner* polynomials define the structure of the inverses of *polynomial-Vandermonde* matrices $V_P(x) = \left[\begin{array}{c} P_{j-1}(x_i) \end{array} \right]$, and use this description to generalize to $V_P(x)$ the Parker-Forney-Traub inversion algorithm. We show that in the case when the polynomials $\{P_k(x)\}$ involved in $V_P(x)$ are the *Szegö polynomials*, the properties of the corresponding *unitary Hessenberg* matrix allow us a dramatic simplification, leading to fast $O(n^2)$ computational procedures for inversion of what we suggest to call Szegö-Vandermonde matrices $V_P(x)$.

1 Introduction

Vandermonde matrices of the form

$$V(x) = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix} \tag{1.1}$$

are classical, and explicit expressions for their determinants and inverses are well-known. The structure in (1.1) can be exploited to design *fast algorithms*. The algebraic complexity of such fast algorithms is typically by an order-of-magnitude less than the one of the standard (structure-ignoring) methods. For example, standard matrix inversion methods require $O(n^3)$ operations. However, the structure of $V(x)^{-1}$ described by Kowalewski in [K32] can

*This work was supported in part by NSF contract CCR 9732355. The views and conclusions contained in these documents are those of the author(s) and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Science Foundation, or the U.S. Government.

†Department of Mathematics and Computer Science, Georgia State University, Atlanta, GA 30303,
E-mail: volshevsky@cs.gsu.edu Web: http://www.cs.gsu.edu/matvro

be used to design a fast $O(n^2)$ algorithm. This fast inversion algorithm has been re-derived in the mathematical and engineering literature several times, and now it is usually associated with the names of Parker [P64], Forney [F66] and Traub [T66] (see also [Wertz65], [K69]).

Classical Vandermonde matrices (1.1) appear in polynomial computations exploiting the monomial basis $\{1, x, x^2, \ldots, x^{n-1}\}$. An alternative use of orthogonal on a real interval polynomials gives rise to the more general *three-term Vandermonde matrices*,

$$V_R(x) = \begin{bmatrix} r_0(x_1) & r_1(x_1) & \cdots & r_{n-1}(x_1) \\ r_0(x_2) & r_1(x_2) & \cdots & r_{n-1}(x_2) \\ \vdots & \vdots & & \vdots \\ r_0(x_n) & r_1(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix}, \tag{1.2}$$

where the polynomials $\{r_0(x), r_1(x), \ldots, r_{n-1}(x)\}$ satisfy three-term recurrence relations. A good performance record of fast algorithms for Vandermonde matrices attracted much attention in the numerical linear algebra literature[1], and the Parker-Forney-Traub algorithm was generalized to invert Chebyshev-Vandermonde matrices [GO94], and three-term Vandermonde matrices [CR93].

Along with the above real-line settings (giving rise to three-term Vandermonde matrices), it is important to consider the case when the inner product is defined on the unit circle, i.e.,

$$< p(x), q(x) >= \frac{1}{2\pi} \int_{-\pi}^{\pi} p(e^{i\theta}) \cdot [q(e^{i\theta})]^* w^2(\theta) d\theta. \tag{1.3}$$

Polynomials $\Phi = \{\phi_k(x)\}$ orthogonal with respect to (1.3) appear in various signal processing applications, and they are called the *Szegö polynomials*. It is well known that the Szegö polynomials are completely described by the two-term recurrence relations [GS58], [G48],

$$\begin{bmatrix} \phi_0^{\#}(x) \\ \phi_0(x) \end{bmatrix} = \frac{1}{\mu_0} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \begin{bmatrix} \phi_{k+1}^{\#}(x) \\ \phi_{k+1}(x) \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} 1 & -\rho_{k+1} \\ -\rho_{k+1}^* & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & x \end{bmatrix} \begin{bmatrix} \phi_k^{\#}(x) \\ \phi_k(x) \end{bmatrix}, \tag{1.4}$$

where the numbers $\{\rho_0, \rho_1, \ldots, \rho_n\}$, are called *reflection coefficients* (the names *parcor coefficients* and *Schur parameters* are also in use). The numbers $\mu_k = \sqrt{1 - |\rho_k|^2}$ are called the *complementary parameters* ($\mu_k := 1$ if $\rho_k = 1$), and $\phi_k^{\#}(x) = x^k[\phi(\frac{1}{x^*})]^*$. The reflection coefficients define the Hessenberg matrix

$$C_\Phi = \begin{bmatrix} -\rho_1\rho_0^* & -\rho_2\mu_1\rho_0^* & -\rho_3\mu_2\mu_1\rho_0^* & \cdots & -\rho_{n-1}\mu_{n-2}\cdots\mu_1\rho_0^* & -\rho_n\mu_{n-1}\cdots\mu_1\rho_0^* \\ \mu_1 & -\rho_2\rho_1^* & -\rho_3\mu_2\rho_1^* & \cdots & -\rho_{n-1}\mu_{n-2}\cdots\mu_2\rho_1^* & -\rho_n\mu_{n-1}\cdots\mu_2\rho_1^* \\ 0 & \mu_2 & -\rho_3\rho_1^* & \cdots & -\rho_{n-1}\mu_{n-2}\cdots\mu_3\rho_2^* & -\rho_n\mu_{n-1}\cdots\mu_3\rho_2^* \\ \vdots & \ddots & \mu_3 & & \vdots & \vdots \\ \vdots & & \ddots & \ddots & -\rho_{n-1}\rho_{n-2}^* & -\rho_n\mu_{n-1}\rho_{n-2}^* \\ 0 & \cdots & \cdots & 0 & \mu_{n-1} & -\rho_n\rho_{n-1}^* \end{bmatrix} \tag{1.5}$$

[1] Interestingly, most of excellent numerical results were reported for the Björck-Pereyra algorithm for solving Vandermonde linear equations (see, e.g., [BP70], [GL89], [BKO00])). In contrast, the fast inversion algorithm for Vandermonde matrices has been incorrectly regarded as inaccurate, mostly because of the poor numerical performance of its version of [T66]. It was numerically demonstrated in [GO96] that the Parker version [P64] typically produces an excellent accuracy.

which has many nice properties, for example, it is well-known that C_Φ differs from unitary only in the last column. Such almost-unitary Hessenberg matrices have been studied in signal processing literature, because they describe the state-space structure for lattice digital filters, see, e.g., [ML80], [KP83], [TKH83], [K85]. The nice structure of C_Φ has been independently studied in numerlical linear algebra literature, and exploiting it allowed to develop efficient algorithms for several problems involving the Szegö polynomials, see [G82], [G86], [GR90], [AGR93], [ACR96] among others. We would like to also mention that in the operator theory literature, see, e.g., [C84], [FF89], the matrices of the form (1.5) are associated with the *Naimark dilation.*

In a parallel paper [O98] we introduce the Horner-Szego polynomials using the language of discrete transmission lines, and then use them for eigenvalue computations. In this paper we use the structure of almost-unitary Hessenberg matrices to study the Horner-Szego polynomials. We then use them to generalize the Parker-Forney-Traub algorithm to what we call Szegö-Vandermonde matrices, i.e., the matrices of the form (1.2) involving the Szegö polynomials.

The paper is structured as follows. In the next section we recall the conventional Parker-Forney-Traub algorithm, providing its new interpretation in terms of the corresponding companion matrices. This interpretation serves as a starting point to formulate in Sec. 3 the generalized Parker-Forney-Traub algorithm for inversion of general polynomial Vandermonde matrices $V_R(x)$, in which polynomials only satisfy $\deg r_k(x) = k$. This algorithm is designed via a simple operation on the corresponding *confederate* matrix (i.e., the Hessenberg matrix generalizing the companion matrix). In section 4 we specify the generalized Parker-Forney-Traub algorithm to Szegö-Vandermonde matrices, for which the corresponding confederate matrix has the form shown in (1.5). This allows to achieve a favorable cost $O(n^2)$. Some conclusions are offered in the final section.

2 Associated (Horner) polynomials and the Parker-Forney-Traub algorithm

2.1. Classical Horner polynomials. Let P be the system of $n+1$ polynomials $P = \{1, x, x^2, ..., x^{n-1}, b(x)\}$, where $b(x) = b_0 + b_1 \cdot x + ... + b_{n-1} \cdot x^{n-1} + b_n \cdot x^n$. Then the first divided difference of $b(x)$ has the Hankel form

$$B(x,y) = \frac{b(x) - b(y)}{x - y} = \sum_{i=1}^{n} b_i \cdot (x^{i-1} + x^{i-2} \cdot y + ... + x \cdot y^{i-2} + x^{i-1}) = \tag{2.1}$$

$$= \begin{bmatrix} 1 & x & x^2 & \cdots & x^{n-1} \end{bmatrix} \cdot \begin{bmatrix} b_1 & b_2 & b_3 & \cdots & b_n \\ b_2 & b_3 & \cdots & b_n & 0 \\ b_3 & \cdots & b_n & \iddots & \vdots \\ \vdots & \iddots & \iddots & & \vdots \\ b_n & 0 & \cdots & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^{n-1} \end{bmatrix} =$$

$$= \left[\begin{array}{ccccc} 1 & x & x^2 & \cdots & x^{n-1} \end{array} \right] \cdot \left[\begin{array}{c} \hat{p}_{n-1}(y) \\ \hat{p}_{n-2}(y) \\ \vdots \\ \hat{p}_1(y) \\ \hat{p}_0(y) \end{array} \right] = \sum_{i=0}^{n-1} x^i \cdot \hat{p}_{n-1-i}(y), \tag{2.2}$$

where the polynomials $\hat{P} = \{\hat{p}_0(x), \hat{p}_1(x), ..., \hat{p}_{n-1}(x),\ \hat{p}_n(x)\}$, with $\hat{p}_n(x) = b(x)$ satisfy the recurrence relations

$$\hat{p}_0(x) = b_n, \qquad \hat{p}_i(x) = x \cdot \hat{p}_{i-1}(x) + b_{n-i}, \tag{2.3}$$

and they are usually coined with the name of Horner, who used them for solving a single nonlinear equation. However, as was noted in [C11], the Horner's method was anticipated by Ruffini, and moreover Horner himself pointed out in [H1819], that these polynomials were known to Lagrange [L1775]. However, all of the above men had been anticipated by Chinese mathematicians, see e.g. [C11]. For these historical reasons in [T66] polynomials $\hat{P} = \{\hat{p}_0(x), \hat{p}_1(x), ..., \hat{p}_{n-1}(x), \hat{p}_n(x)\}$ were called in [T66] the *associated* with $P = \{1, x, x^2, ..., x^{n-1}, b(x)\}$ polynomials.

2.2. The Parker-Forney-Traub inversion algorithm. Traub used the simplicity of the recursion (2.3) to derive in [T66] a fast $O(n^2)$ algorithm for inversion of Vandermonde matrices. Before describing his result, let us introduce the necessary notations. Denote by

$$V_R(x) = \left[\begin{array}{cccc} r_0(x_1) & r_1(x_1) & \cdots & r_{n-1}(x_1) \\ r_0(x_2) & r_1(x_2) & \cdots & r_{n-1}(x_2) \\ \vdots & \vdots & & \vdots \\ r_0(x_n) & r_1(x_n) & \cdots & r_{n-1}(x_n) \end{array} \right] \tag{2.4}$$

a polynomial Vandermonde matrix corresponding to the set of nodes $x = (x_1, x_2, ..., x_n)$ and to the system of polynomials $R = \{r_0(x), r_1(x), ..., r_n(x)\}$. Let $\tilde{I}$ be the antidiagonal identity matrix, and superscript ^{T} means transpose. With these notations the equality (2.2) immediately implies

$$V_P(x) \cdot \tilde{I} \cdot V_{\hat{P}}(x)^T = \left[\begin{array}{c} \frac{b(x_i)-b(x_j)}{x_i-x_j} \end{array} \right]_{1 \le i,j \le n}, \tag{2.5}$$

where the diagonal entries are understood as $b'(x_i)$ ($i = 1, 2, ..., n$). Observe that (2.5) is valid for arbitrary polynomial $b(x)$, and if we additionally set

$$b(x) = \Pi_{i=1}^n (x - x_i), \tag{2.6}$$

then

$$V_P(x)^{-1} = \tilde{I} \cdot V_{\hat{P}}^T \cdot \mathrm{diag}(c_1, c_2, ..., c_n), \tag{2.7}$$

where

$$c_i = b'(x_i) = \frac{1}{\Pi_{\substack{k=1 \\ k \neq i}}^n (x_k - x_i)}. \tag{2.8}$$

Traub used (2.7) to formulate the following efficient inversion algorithm.

1). Compute the numbers c_i by (2.8).

2). Compute the coefficients b_i ($i = 0, 1, ..., n$) of the polynomial $b(x)$ in (2.6).

3). Compute the entries of $V_{\hat{P}}(x)$ using recursion (2.3).

4). Multiply the matrices on the right hand side of (2.7).

These computations require performing only $6n^2$ arithmetic operations, achieving a favorable efficiency as compared to the expensive complexity $O(n^3)$ of the Gaussian elimination. A key point that makes this speed-up possible is that for the monomial system P, the recurrence relations (2.3) for the Horner polynomials $\hat{P}$ remain sparse. Our next goal is to carry over the Parker-Forney-Traub algorithm to the more general polynomial-Vandermonde matrices $V_R(x)$; to this end we need to define *generalized Horner polynomials* for an arbitrary polynomial system R, and to obtain for them an analogue of (2.3). This will be achieved by using a useful matrix interpretation of (2.3), given next.

2.3. A confederate matrix interpretation. Let polynomials $R = \{r_0(x),\ r_1(x)\ , ...,\ r_n(x)\}$ be specified by the recurrence relations

$$r_k(x) = \alpha_k \cdot x \cdot r_{k-1}(x) - a_{k-1,k} \cdot r_{k-1}(x) - a_{k-2,k} \cdot r_{k-2}(x) - ... - a_{0,k} \cdot r_0(x). \tag{2.9}$$

Following [MB79], define for the polynomial

$$b(x) = b_0 \cdot r_0(x) + b_1 \cdot r_1(x) + ... + b_{n-1} \cdot r_{n-1}(x) + b_n \cdot r_n(x) \tag{2.10}$$

its *confederate* matrix

$$C_R(b) = \begin{bmatrix} \frac{a_{01}}{\alpha_1} & \frac{a_{02}}{\alpha_2} & \frac{a_{03}}{\alpha_3} & \cdots & \cdots & \frac{a_{0,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_0}{b_n} \\ \frac{1}{\alpha_1} & \frac{a_{12}}{\alpha_2} & \frac{a_{13}}{\alpha_3} & \cdots & \cdots & \frac{a_{1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_1}{b_n} \\ 0 & \frac{1}{\alpha_2} & \frac{a_{23}}{\alpha_3} & \cdots & \cdots & \frac{a_{2,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_2}{b_n} \\ 0 & 0 & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \frac{1}{\alpha_{n-1}} & \frac{a_{n-1,n}}{\alpha_n} - \frac{1}{\alpha_n} \cdot \frac{b_{n-1}}{b_n} \end{bmatrix} \tag{2.11}$$

with respect to the polynomial system R. We refer to [MB79] for many useful properties of the confederate matrix and only recall here that $\det(xI - C_R(b)) = b(x)/(\alpha_0 \cdot \alpha_1 \cdot ... \cdot \alpha_n \cdot b_n)$, and that similarly, the characteristic polynomial of the $k \times k$ leading submatrix of $C_R(b)$ is equal to $r_k(x)/\alpha_0 \cdot \alpha_1 \cdot ... \cdot \alpha_k$.

In the simplest case $P = \{1, x, x^2, ..., x^{n-1}, b(x)\}$, the confederate matrix $C_P(b)$ reduces to the well known companion matrix

$$C_P(b) = \begin{bmatrix} 0 & 0 & \cdots & 0 & -b_0 \\ 1 & 0 & \cdots & 0 & -b_1 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -b_{n-1} \end{bmatrix} \tag{2.12}$$

of the monic polynomial $b(x) = b_0 + b_1 \cdot x + ... + b_{n-1} \cdot x^{n-1} + x^n$. Now observe that the recurrence relations (2.3) for the Horner polynomials $\hat{P} = \{\hat{p}_0(x), ..., \hat{p}_n(x)\}$ give the following nice structure for the corresponding confederate matrix :

$$C_{\hat{P}}(\hat{p}_n) = \begin{bmatrix} -b_{n-1} & -b_{n-2} & \cdots & -b_1 & -b_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & 0 \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}. \tag{2.13}$$

Equalities (2.12) and (2.13) imply the following statement.

Proposition 2.1 *The confederate matrix of the monomials $P = \{1, x, x^2, ..., x^{n-1}, b(x)\}$ and that of the Horner polynomials $\hat{P} = \{\hat{p}_0(x), ..., \hat{p}_n(x)\}$ are related by*

$$C_{\hat{P}}(\hat{p}_n) = \tilde{I} \cdot C_P(b)^T \cdot \tilde{I} \tag{2.14}$$

where $\tilde{I}$ is the antidiagonal identity matrix.

In the next section we shall use (2.14) to extend the concept of the Horner polynomials to arbitrary polynomial system R.

2.4. Generalized Horner polynomials and displacement structure. In [KO94] we used (for arbitrary given polynomials $R = \{r_0(x), .., r_n(x)\}$) the *generalized Horner* polynomials $\hat{R} = \{\hat{r}_0(x), ..., \hat{r}_n(x)\}$ that were specified by their confederate matrix

$$C_{\hat{R}}(\hat{r}_n) = \tilde{I} \cdot C_R(r_n)^T \cdot \tilde{I}, \qquad (\text{with} \quad \hat{r}_n(x) = r_n(x)). \tag{2.15}$$

Such polynomials $\hat{R}$ have been found to be useful to describe the structure of the inverses of the more general *polynomial Vandermonde-like matrices* S that are defined as having low *displacement rank*

$$\text{rank}(\text{diag}(x_1, ..., x_n) \cdot S - S \cdot C_R(r_n))$$

The motivation for the above definition can be inferred from the easily verified fact that the displacement rank of $V_R(x)$ is one :

$$\text{diag}(x_1, ..., x_n) \cdot V_R(x) - V_R(x) \cdot C_R(r_n) = \begin{bmatrix} r_n(x_1) \\ \vdots \\ r_n(x_n) \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 & \frac{1}{\alpha_n} \end{bmatrix}. \tag{2.16}$$

It can be seen that the displacement rank of a matrix is essentially inherited under inversion :

$$\text{rank}(\text{diag}(x_1, ..., x_n) \cdot S - S \cdot C_R(r_n)) = \text{rank}(\text{diag}(x_1, ..., x_n) \cdot (S^{-T} \cdot \tilde{I}) - (S^{-T} \cdot \tilde{I}) \cdot C_{\hat{R}}(r_n)).$$

Briefly, the passage from S to $(S^{-T} \cdot \tilde{I})$ corresponds to the passage from polynomials R to the generalized Horner polynomials $\hat{R}$. This fact allowed us to to obtain in [KO94] explicit inversion formulas for polynomial Vandermonde-*like* matrices in terms of generalized Horner polynomials.

3 The generalized Parker-Forney-Traub algorithm

In this section we show how the generalized Horner polynomials $\hat{R}$ define the structure of $V_R(x)^{-1}$. To this end we an auxiliary result, given next.

3.1. Change of basis. Along with a polynomial system $Q = \{q_0(x), q_1(x)..., q_n(x)\}$ (satisfying $\deg q_k(x) = k$) consider another system $R = \{r_0(x), r_1(x), ...r_n(x)\}$ with $r_n(x) = q_n(x)$ and

$$r_k(x) = s_{0,k} \cdot q_0(x) + s_{1,k} \cdot q_1(x) + ... + s_{k,k} \cdot q_k(x), \qquad s_{k,k} \neq 0. \tag{3.1}$$

Clearly,

$$\left[\begin{array}{ccccc} q_0(x) & q_1(x) & q_2(x) & \cdots & q_{n-1}(x) \end{array}\right] \cdot S_{QR} = \left[\begin{array}{ccccc} r_0(x) & r_1(x) & r_2(x) & \cdots & r_{n-1}(x) \end{array}\right], \tag{3.2}$$

and

$$V_Q(x) \cdot S_{QR} = V_R(x), \tag{3.3}$$

where S_{QR} is an upper triangular matrix whose columns are formed from the coefficients of the polynomials in (3.1). According to [MB79] we have

$$C_R(r_n) = S_{QR}^{-1} \cdot C_Q(q_n) \cdot S_{QR}. \tag{3.4}$$

From here and (2.15) it follows immediately that

$$C_{\hat{R}}(\hat{r}_n) = (\tilde{I} S_{QR}^{-T} \tilde{I})^{-1} \cdot C_{\hat{Q}}(\hat{q}_n) \cdot (\tilde{I} S_{QR}^{-T} \tilde{I}). \tag{3.5}$$

Briefly, the similarity matrices for the four polynomial systems $P, \hat{P}, R, \hat{R}$ can be depicted as follows ;

$$\begin{array}{ccc} P & \xrightarrow{S_{PR}} & R \\ \downarrow & & \downarrow \\ \hat{P} & \xrightarrow{\tilde{I} \cdot S_{PR}^{-T} \cdot \tilde{I}} & \hat{R} \end{array} \tag{3.6}$$

3.2. Inversion formula. Therefore (3.3) and(3.6) allow us to deduce from (2.7) the more general formula for the inverse of an arbitrary polynomial-Vandermonde matrix :

$$V_R(x)^{-1} = \tilde{I} \cdot V_{\hat{R}}^T \cdot \operatorname{diag}(c_1, ..., c_n), \tag{3.7}$$

where c_k are given by (2.8). This formula allows us to extend the Parker-Forney-Traub algorithm to invert an arbitrary polynomial-Vandermonde matrix $V_R(x)$.

3.3. Inversion algorithm. To invert $V_R(x)$ by (3.7) all we need is

(a) to compute, for $R = \{r_0(x), ..., r_{n-1}(x), b(x)\}$ with $b(x) = \prod_{k=1}^{n}(x - x_k)$ the matrix $C_R(b)$. To this end observe that the matrix $C_R(b)$ does not depend upon the last polynomial in R, and in particular $C_R(b) = C_{\bar{R}}(b)$, where $\bar{R} = \{r_0(x), ..., r_{n-1}(x), x \cdot r_{n-1}(x)\}$. Thus to obtain $C_R(b)$ we need to find the coefficients of $b(x)$ decomposed in the basis $\bar{R}$, which can be done recursively by starting with $r_n^{(0)}(x) = 1$ and by updating $r_n^{(k+1)}(x) = (x - x_{k+1}) \cdot r_n^{(k)}(x)$,

(b) to compute, using (2.15) the entries of $V_{\hat{R}}(x)$.

A convenient procedures for doing these two are provided in the next two simple lemmas.

Lemma 3.1 *Let $R = \{r_0(x), ..., r_n(x)\}$ be given by (2.9), 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*

$$\begin{bmatrix} b_0 \\ \vdots \\ b_{k+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \left[\begin{array}{cccc|c} \multicolumn{4}{c|}{C_R(r_n)} & 0 \\ \hline 0 & \cdots & 0 & \frac{1}{\alpha_n} & 0 \end{array}\right] \begin{bmatrix} a_0 \\ \vdots \\ a_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.8}$$

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 (3.8).

Lemma 3.2 *Let $R = \{r_0(x), ..., r_n(x)\}$ be given by (2.9), Then (2.15) translates into the following recursion for the generalized Horer polynomials*

$$\hat{R}_0(x) = \hat{\alpha}_0,$$

$$\hat{R}_k(x) = \hat{\alpha}_k \cdot x \cdot \hat{R}_{k-1}(x) - \hat{a}_{0,k} \cdot \hat{R}_0(x) - \hat{a}_{1,k} \cdot \hat{R}_1(x) - ... - \hat{a}_{k-1,k} \cdot \hat{R}_{k-1}(x), \tag{3.9}$$

where

$$\hat{\alpha}_k = \alpha_{n-k}, \qquad (k = 0, 1, ..., n),$$

and

$$\hat{a}_{k,j} = \frac{\alpha_{n-j}}{\alpha_{n-k}} a_{n-j,n-k} \qquad (k = 0, 1, ...n-1; j = 1, 2, ..., n). \tag{3.10}$$

These two auxiliary statements allows us to write down the following algorithm.

The generalized Parker-Forney-Traub algorithm.

INPUT : n numbers $x = (x_1, ..., x_n)$ and n polynomials $\{r_0(x), r_1(x), ..., r_{n-1}(x)\}$, specified by the coefficients of the recurrence relations (2.9).

OUTPUT : The entries of $V_P(x)^{-1}$.

STEPS :

1). Compute the numbers c_i by (2.8).

2). Compute the coefficients in $b(x) = \prod_{k=1}^{n}(x - x_k) = \alpha_n \cdot x \cdot p_{n-1}(x) - a_{n-1,n} \cdot p_{n-1}(x) - ... - a_{0,n} \cdot p_0(x)$ by

(2).1)) Set $\left[\begin{array}{cccc} -a_0^{(0)} & \cdots & -a_{n-1}^{(0)} & \alpha_n^{(0)} \end{array}\right] = \left[\begin{array}{cccc} \frac{1}{\alpha_0} & 0 & \cdots & 0 \end{array}\right]$

(2).2)) For $k = 1 : n$ do

$$\left[\begin{array}{c} -a_0^{(k)} \\ \vdots \\ -a_{n-1}^{(k)} \\ \alpha_n^{(k)} \end{array}\right] = \left(\left[\begin{array}{c|c} C_{\bar{R}}(x \cdot r_{n-1}(x)) & 0 \\ \hline 0 \ \cdots \ 0 \ 1 & 0 \end{array}\right] - x_k \cdot I \right) \cdot \left[\begin{array}{c} -a_0^{(k-1)} \\ \vdots \\ -a_{n-1}^{(k-1)} \\ \alpha_n^{(k-1)} \end{array}\right]$$

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

3). Compute the entries of $V_{\hat{P}}(x)$ using recursion (3.9).

4). Multiply the matrices on the right hand side of (2.7).

In the case of the monomial basis, $R = P$, this procedure reduces to the conventional Parker-Forney-Traub algorithm, described in Sec. 1. For polynomials orthogonal on a real interval, this procedure reduces to the Calvetti-Reichel algorithm [CR93] for inversion of three-term Vandermonde matrices. In both cases the computational complexity is $O(n^2)$ operations, however for general polynomials, satisfying only $\deg p_k(x) = k$, the above generalized Parker-Forney-Traub algorithm requires the same amount $O(n^3)$ operations as standard methods. In the next section we show that in another important special case of Szegö polynomials our inversion algorithm also admits a simplification, resulting in an order-of-magnitude reduction in the computational complexity.

4 Inversion of Szegö-Vandermonde matrices

4.1. Computational complexity. As it has been just mentioned, although in the general case the generalized Parker-Forney-Traub algorithm requires $O(n^3)$ operations, it has a favorably low complexity of $O(n^2)$ operations for the monomials and for polynomials orthogonal on a real interval. As was just noted, the crucial point that makes this speed-up possible is the sparsity of the corresponding confederate matrices $C_R(r_n)$. Clearly, the sparsity of $C_R(r_n)$ means that the n^2 entries of $V_R(x)$ are completely defined by a smaller number $O(n)$ of parameters. Therefore it is not surprising that operating on these parameters we are able to achieve a faster computation. However in a general situation the confederate matrix also involves $O(n^2)$ parameters, so the generalized algorithms proposed above have the same complexity $O(n^3)$ operations as structure-ignoring algorithms. In the rest of the paper we consider another important case where R are the Szegö polynomials, and show that in this case our algorithms also allow an efficient simplification.

4.2. Recurrence relations for Szegö polynomials. Let $\Phi = \{\phi_0(x), ..., \phi_n(x)\}$ denote the family of Szegö polynomials, i.e., those satisfying the two-term recurrence relations

$$\left[\begin{array}{c} \phi_0(x) \\ \phi_0^{\#}(x) \end{array}\right] = \frac{1}{\mu_0} \left[\begin{array}{c} 1 \\ 1 \end{array}\right],$$

$$\left[\begin{array}{c} \phi_{i+1}(x) \\ \phi_{i+1}^{\#}(x) \end{array}\right] = \frac{1}{\mu_{i+1}} \left[\begin{array}{cc} 1 & -\rho_{i+1} \\ -\rho_{i+1}^* & 1 \end{array}\right] \left[\begin{array}{cc} x & 0 \\ 0 & 1 \end{array}\right] \left[\begin{array}{c} \phi_i(x) \\ \phi_i^{\#}(x) \end{array}\right], \tag{4.1}$$

where the numbers $\{\rho_1, \rho_2, \ ..., \ \rho_n\}$ are called the *reflection coefficients* (the names Schur parameters, parcor coefficients are also in use), and the numbers $\mu_k = \sqrt{1-|\rho_k|^2}$ are called *complementary* parameters (if $|\rho_n| = 1$ then we set $\mu_n := 1$ for consistency). The Szegö polynomials are orthonormal with respect to a suitable inner product on the unit circle,

$$< p(x), q(x) >= \int_{-\pi}^{\pi} p(e^{i\theta}) \cdot [q(e^{i\theta})]^* w^2(\theta) \frac{d\theta}{2\pi},$$

and they are of a particular importance in signal processing applications.

In the rest of the paper we shall specialize our algorithms to $V_\Phi(x)$, which we suggest to call the *Szegö-Vandermonde* matrices.

To efficiently specialize the generalized Parker-Forney-Traub algorithm for $V_\Phi(x)$ we need to write down the corresponding confederate matrix, for this purposes it will be more convenient to use not (4.1), but recently somewhat ignored three-term recurrence relations for Szegö polynomials (see, e.g., [G48])

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

$$\phi_k(x) = [\frac{1}{\mu_k} \cdot x + \frac{\rho_k}{\rho_{k-1}} \frac{1}{\mu_k}] \cdot \phi_{k-1}(x) - \frac{\rho_k}{\rho_{k-1}} \frac{\mu_{k-1}}{\mu_k} \cdot x \cdot \phi_{k-2}(x). \tag{4.2}$$

In fact, when Szegö first introduced the polynomials orthogonal on the unit circle, he gave the expected three-term recursion (4.2). The two-term recursions (4.1) were only recognized about twenty years later and independently by Szegö (1939) and Geronimus (1948).

A straightforward computation easily allows one to convert (4.2) into the n-term recurrence relations of the form (2.9), which describe the corresponding confederate matrix. In fact, the following statement holds.

Lemma 4.1 *The Szegö polynomials in (4.1) (or, equivalently, in (4.2)) satisfy*

$$\phi_0(x) = 1, \qquad \phi_1(x) = \frac{1}{\mu_1}(x \cdot \phi_0(x) + \rho_1 \rho_0^* \cdot \phi_0(x)),$$

where $\rho_0 = -1$, *and*

$$\phi_k(x) = \frac{1}{\mu_k}[x \cdot \phi_{k-1}(x) + \rho_k \ \rho_{k-1}^* \cdot \phi_{k-1}(x) + \rho_k \mu_{k-1} \rho_{k-2}^* \cdot \phi_{k-2} + \rho_k \mu_{k-1} \mu_{k-2} \rho_{k-3}^* + ...$$

$$... + \rho_k \mu_{k-1} \mu_{k-2} \cdot ... \cdot \mu_2 \rho_1^* \cdot \phi_1(x) + \rho_k \mu_{k-1} \mu_{k-2} \cdot ... \cdot \mu_1 \cdot \phi_0(x)], \tag{4.3}$$

or, equivalently, the confederate matrix (2.11) for $b(x) = b_n \cdot \phi_n(x) + ... + b_0 \cdot \phi_0(x)$ *is of the form*

$$C_\Phi(b) = \begin{bmatrix} -\rho_1\rho_0^* & -\rho_2\mu_1\rho_0^* & -\rho_3\mu_2\mu_1\rho_0^* & \cdots & -\rho_{n-1}\mu_{n-2}...\mu_1\rho_0^* & -\rho_n\mu_{n-1}...\mu_1\rho_0^* - \frac{b_0}{b_n}\mu_n \\ \mu_1 & -\rho_2\rho_1^* & -\rho_3\mu_2\rho_1^* & \cdots & -\rho_{n-1}\mu_{n-2}...\mu_2\rho_1^* & -\rho_n\mu_{n-1}...\mu_2\rho_1^* - \frac{b_1}{b_n}\mu_n \\ 0 & \mu_2 & -\rho_3\rho_1^* & \cdots & -\rho_{n-1}\mu_{n-2}...\mu_3\rho_2^* & -\rho_n\mu_{n-1}...\mu_3\rho_2^* - \frac{b_2}{b_n}\mu_n \\ \vdots & \ddots & \mu_3 & & \vdots & \vdots \\ \vdots & & \ddots & \ddots & -\rho_{n-1}\rho_{n-2}^* & -\rho_n\mu_{n-1}\rho_{n-2}^* - \frac{b_{n-2}}{b_n}\mu_n \\ 0 & \cdots & \cdots & 0 & \mu_{n-1} & -\rho_n\rho_{n-1}^* - \frac{b_{n-1}}{b_n}\mu_n \end{bmatrix} \tag{4.4}$$

The Hessenberg matrix (4.4) differs from the unitary only in the last column, and this matrix was written down [G82]) (in a slightly different form), and [KP83]. The useful properties of this unitary Hessenberg matrices were later intensively studied in the signal processing (where it appears as a system matrix of orthogonal lattice filters), and quite independently in numerical analysis literature. Below we exploit a simple structure of $C_\Phi(b)$ to speed-up the computations with $V_\Phi(x)$.

4.3. Multiplication of $C_\Phi(\phi_n)$ by a vector. Recall that in order to specify the generalized Parker-Forney-Traub algorithm to the Szegö-Vandermonde matrices we need to simplify the computations in the steps 2 and 3 of this algorithm, based on lemmas 2.1 and 2.2, respectively (cf. with the discussion preceding Lemma 2.1). Specifically, we need

(a) an efficient procedure for multiplication of $C_\Phi(\phi_n)$ by a vector, used in (3.8) to compute the coefficients of $b(x) = \prod_{i=1}^{n}(x - x_i)$ with respect to the polynomial basis Φ.

(b) an analog of (2.3), i.e., a computationally efficient recursion for the successive evaluation of the generalized Horner polynomials
$\hat{R}$ associated with the $R = \{\phi_0(x), \phi_1(x), ..., \phi_{n-1}(x), b(x)\}$.

In fact, the first task in **(a)** is easily achieved by using the well-known decomposition

$$C_\Phi(\phi_n) = H(\rho_1) \cdot H(\rho_2) \cdot ... \cdot H(\rho_{n-1}) \cdot \tilde{H}(\rho_n). \tag{4.5}$$

into the product of the unitary matrices of the form $H(\rho_k) = \operatorname{diag}\{I_{k-1}, \begin{bmatrix} \rho_k & \mu_k \\ \mu_k & -\rho_k^* \end{bmatrix}, I_{n-k-1}\}$ for $k = 1, 2, ..., n-1$, and $\tilde{H}_n = \operatorname{diag}\{I_{n-1}, \rho_n\}$. Therefore, (4.5) reduces multiplying $C_\Phi(\phi_n)$ by a vector to $n-1$ circular rotations, thus suggesting an efficient implementation for the step 2 of the generalized Parker-Forney-Traub algorithm of section 2 for Szegö-Vandermonde matrices.

4.4. Generalized Horner polynomials. To address the second task in (b), we first observe that if $\rho_n = -1$ (and, for consistency, $\mu_n = 1$), then the generalized Horner polynomials $\hat{\Phi} = \{\hat{\phi}_0(x), ..., \hat{\phi}_n(x)\}$ associated with $\Phi = \{\phi_0(x), ..., \phi_n(x)\}$ are themselves the Szegö polynomials defined by their reflection coefficients

$$\{\hat{\rho}_0, \hat{\rho}_1, ..., \hat{\rho}_n\} = \{\rho_n^*, \rho_{n-1}^*, ..., \rho_1^*, \rho_0^*\}, \tag{4.6}$$

(now it is clear why we adopted earlier the convention $\rho_0 = -1$). Briefly, in this simplest case the operation of the passage to $\hat{\Phi}$ is reduced to the reversion and complex conjugation for the reflection coefficients. However, for $R = \{\phi_0(x), ..., \phi_{n-1}(x), b(x)\}$ the generalized Horner polynomials $\hat{R}$ are not the Szegö polynomials, and moreover, their confederate matrix,

$$C_{\hat{R}}(b) = \tag{4.7}$$

$$\begin{bmatrix} -\rho_{n-1}^*\rho_n - \frac{b_{n-1}}{b_n}\hat{\mu_0} & -\rho_{n-2}\mu_{n-1}\rho_n - \frac{b_{n-2}}{b_n}\hat{\mu_0} & \cdots & -\rho_1^*\mu_2...\mu_{n-1}\rho_n - \frac{b_1}{b_n}\hat{\mu_0} & \rho_0^*\mu_1...\mu_{n-1}\rho_n - \frac{b_0}{b_n}\hat{\mu_0} \\ \mu_{n-1} & -\rho_{n-2}^*\rho_{n-1} & \cdots & -\rho_1^*\mu_2...\mu_{n-2}\rho_{n-1} & -\rho_0^*\mu_1...\mu_{n-2}\rho_{n-1} \\ 0 & \mu_{n-2} & & \vdots & \vdots \\ \vdots & & \ddots & -\rho_1^*\rho_2 & -\rho_0^*\mu_1\rho_2 \\ 0 & \cdots & 0 & \mu_1 & -\rho_0^*\rho_1 \end{bmatrix}$$

is not sparse, which means that we have the n-term recurrence relations for $\hat{R}$. However, the arguments converse to those used for deducing (4.3) from (4.2) allows to obtain the following simpler recursion.

Lemma 4.2 *Let the polynomial $b(x)$ be arbitrary, and the $\{\phi_0(x), ..., \phi_n(x)\}$ be the first $n+1$ Szegö polynomials, specified by $\{\rho_1, ..., \rho_n\}$ via (4.2), and let $\rho_0 = -1$. Then the generalized Horner polynomials $\hat{R} = \{\hat{r}_0(x), ..., \hat{r}_n(x)\}$ associated with $R = \{\phi_0(x), ..., \phi_{n-1}(x), b(x)\}$ are given by*

1. Three-term recurrence relations.

$$\hat{r}_0(x) = 1, \qquad \hat{r}_1(x) = \{\frac{1}{\hat{\mu}_1} \cdot x\hat{r}_0(x) - \frac{\hat{\rho}_1\hat{\rho}_0^*}{\hat{\mu}_1}\hat{r}_0(x)\} - \frac{b_{n-1}}{b_n}\hat{\mu}_0.$$

$$\hat{r}_k(x) = \{[\frac{1}{\hat{\mu}_k}\cdot x + \frac{\hat{\rho}_k}{\hat{\rho}_{k-1}}\frac{1}{\hat{\mu}_k}]\hat{r}_{k-1}(x) - \frac{\hat{\rho}_k}{\hat{\rho}_{k-1}}\frac{\hat{\mu}_{k-1}}{\hat{\mu}_k}\cdot x\cdot\hat{r}_{k-2}(x)\} - \frac{b_{n-k} - b_{n-k+1}\hat{\mu}_{k-1}\frac{\hat{\rho}_k}{\hat{\rho}_{k-1}}}{b_n \cdot \hat{\mu}_k}\hat{\mu}_0\hat{r}_0(x). \tag{4.8}$$

where $\hat{\rho}_k = \rho_{n-k}$ for $k = 0, 1, ..., n$ and $\hat{\mu}_k = \sqrt{1 - |\hat{\rho}_k|^2}$, $\hat{\mu}_n = 1$ (and if $|\hat{\rho}_n| = 1$, then $\hat{\mu}_n := 1$).

2. Two-term recurrence relations.

$$\begin{bmatrix} \tilde{\phi}_0(x) \\ \hat{\phi}_0^{\#}(x) \end{bmatrix} = \frac{1}{\tilde{\mu}_0}\begin{bmatrix} -\tilde{\rho}_0 b_n \\ b_n \end{bmatrix}, \qquad \begin{bmatrix} \tilde{\phi}_k(x) \\ \hat{\phi}_k^{\#}(x) \end{bmatrix} = \frac{1}{\tilde{\mu}_k}\begin{bmatrix} 1 & -\tilde{\rho}_k^* \\ -\tilde{\rho}_k & 1 \end{bmatrix}\begin{bmatrix} \tilde{\phi}_{k-1}(x) \\ x\hat{\phi}_{k-1}^{\#}(x) + b_{n-k} \end{bmatrix}, \tag{4.9}$$

In a parallel paper [O98] we described the Horner-Szego polynomials and gave an interpretation of the formulas (4.8) of (4.9) using the language of discrete transmission lines (and used them for eigenvalue computations). Here we use a purely algebraic technique based on almost-unitary Hessenberg matrices, and apply it to derive fast algorithms similar to the one of Parker-Forner-Traub.

Either formula (4.8) of (4.9) suggests an efficient implementation for the step 3 of the generalized Parker-Forney-Traub algorithm of section 2 for Szegö-Vandermonde matrices. Summarizing, the overall complexity of the suggested inversion algorithm is $O(n^2)$.

5 Conclusion

In this paper we generalized the well-known Parker-Forney-Traub algorithm to polynomial-Vandermonde matrices. The algorithm is derived by exploiting the properties of the corresponding confederate matrix (i.e., Hessenberg matrix capturing the recurrence relations). In the important case of Szegö polynomials the corresponding Hessenberg matrix differs from unitary only in the last column. This nice property is exploited to obtain a dramatic simplification of the new algorithms, leading to a favorably small computational complexity $O(n^2)$ operations, as opposed to the usual $O(n^3)$ of standard (structure-ignoring) methods.

References

[ACR96] G.Ammar, D.Calvetti and L.Reichel, *Continuation methods for the computation of zeros of Szegö polynomials*, Linear Algebra and Its Applications, **249** (1996), 125-155.

[AGR93] G.Ammar, W.Gragg and L.Reichel, *An analogue for the Szegö polynomials of the Clenshaw algorithm*, J. Computational Appl. Math., 46 (1993) pp., 211-216.

[B75] S.Barnett, *A companion matrix analogue for orthogonal polynomials*, Linear Algebra Appl., **12** (1975), 197-208.

[BKO00] T.Boros, T.Kailath and V.Olshevsky, *The Fast Björck-Pereyra-type algorithm for solving Cauchy linear equations*, to appear in Linear Algebra and Its Applications, 2000.
Stanford ISL research report, 1995.

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

[C11] F.Cajori, *Horner's method of approximation anticipated by Ruffini*, Bull. Amer. Math. Soc., **17**(1911), 301-312.

[C55] C.Clenshaw, *A note on summation of Chebyshev series*, M.T.A.C., **9**(51)(1955), 118-120.

[C84] T.Constantinescu, *On the structure of the Naimark dilation*, J. of Operator Theory, **12**: 159-175 (1984).

[F66] G.Forney, *Concatenated codes*, The M.I.T. Press, 1966, Cambridge.

[FF89] C.Foias and A.E.Frazho, *The Commutant Lifting Approach to Interpolation Problems*, Birkhauser Verlag, 1989.

[CR93] Calvetti, D. and Reichel, L. : *Fast inversion of Vandermonde-like matrices involving orthogonal polynomials*, BIT, 1993.

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

[G86] W.B.Gragg, *The QR algorithm for unitary Hessenberg matrices*, J.Comput. Appl. Math., **16** (1986), 1-8.

[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.

[GL89] G.Golub and C.Van Loan, *Matrix Analysis*, second edition, John Hopkins U. P., Baltimore, 1989.

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

[GO96] 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, p.205-221.

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

[GR90] W.B.Gragg and L.Reichel, *A divide and conquer method for unitary and orthogonal eigenproblems*, Numer. Math., **57** (1990), 695-718.

[H1819] W.G.Horner, *A new method of solving numerical equations of all orders by continuous approximation*, Philos. Trans. Roy. Soc. London, (1819), 308-335.

[K69] I.Kaufman, *The inversion of the Vandermonde matrix and the transformation to the Jordan canonical form*, IEEE Trans. on Automatic Control, **14**(1969), 774 - 777.

[K85] H.Kimura, *Generalized Schwartz Form and Lattice-Ladder Realizations for Digital Filters*, IEEE Transactions on Circuits and Systems, **32**, No 11 (1985), 1130-1139.

[K32] G. Kowalewski, *Interpolation und genäherte Quadratur*, Teubner, Berlin, 1932.

[KO94] T.Kailath and V.Olshevsky, *Displacement structure approach to polynomial Vandermonde and related matrices*, preprint, 1994.

[KP83] T.Kailath and B.Porat, *State-space generators for orthogonal polynomials*, in Prediction theory and harmonic analysis, The Pesi Masani Volume, V.Mandrekar and H.Salehi (eds.), pp.131-163, North-Holland Publishing Company, 1983.

[L1775] J.L.Lagrange, *Sur les suites recurrentes*, Nouveau Mémories de l'Academie Royale de Berlin, **6**(1775), 183-195.

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

[ML80] M.Morf and D.T.Lee, *State-space structure of ladder canonical forms*, Proc. 18th Conf. on Control and Design, Dec. 1980, 1221-1224.

[O98] 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.

[P64] F.Parker, *Inverses of Vandermonde matrices*, Amer. Math. Monthly, 71 (1964), 410 - 411.

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

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

[SB80] J.Stoer and R.Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1980.

[T66] J. Traub, *Associated polynomials and uniform methods for the solution of linear problems*, SIAM Review, 8, No. 3 (1966), 277 – 301.

[TKH83] M.Takizawa, Hisao Kishi and N.Hamada, *Synthesis of Lattice Digital Filter by the State Space Variable Method*, Trans. IECE Japan, vol. J65-A (1983), 363-370.

[Wertz65] H.J.Wertz, *On the numerical inversion of a recurrent problem : the Vandermonde matrix*, IEEE Trans. on Automatic Control, AC-10, **4**(1965), 492.