Fast Triangular Factorization of the Sum of Quasi-Toeplitz and Quasi-Hankel Matrices*

Ali H. Sayed, Hanoch Lev-Ari,[‡] and Thomas Kailath

Information Systems Laboratory
Stanford University
Stanford, California 94305

ABSTRACT

The literature contains several recent fast algorithms for the triangular factorization of strongly regular Toeplitz-plus-Hankel matrices. In this paper we study the rather more general sum of quasi-Toeplitz and quasi-Hankel matrices, both Hermitian and non-Hermitian. Quasi-Toeplitz and quasi-Hankel matrices are those that are congruent to Toeplitz and Hankel matrices in a special sense. The derivation is based on the concept of displacement structure and its intimate relation to the Schur reduction procedure for triangular factorization. Various special cases covering displacement ranks from two to eight are considered. Several other problems (e.g., factorization of the inverse matrix, solution of exact or overdetermined linear systems) can be reduced to the direct factorization problem.

---

1. INTRODUCTION

Many signal processing problems require the solution of a linear system of equations of large dimensions. In many applications, (e.g., linear prediction, discrete inverse scattering, cascade synthesis, digital filtering, and the partial realization problem), Toeplitz, quasi-Toeplitz, Hankel, or quasi-Hankel matrices need to be factored or inverted. In many cases the coefficient matrix can be represented as the sum of quasi-Toeplitz and quasi-Hankel matrices. For example, in 1955 Gelfand and Levitan [1] reduced the solution of the inverse scattering problem to the solution of a set of linear integral equations. The discretized version of these equations has a Toeplitz-plus-Hankel coefficient matrix [2]. Such coefficient matrices also arise in the design of least-squares linear-phase prediction and smoothing filters [3] and in the least-squares minimization of forward and backward prediction error energies in AR spectral estimation [4, 5]. Also, in digital signal processing, one is often faced with the problem of computing the output noise variance due to fixed-point quantization or input signal quantization. This problem was shown [6] to reduce to the solution of a linear system of equations whose coefficient matrix is the sum of a Toeplitz matrix and a quasi-Hankel matrix.

Several fast inversion algorithms [$O(n^2)$] for Toeplitz-plus-Hankel matrices have been published in recent years. Friedlander et al. used the generalized Levinson-Szegö algorithm [7] and the multichannel Levinson algorithm [8] to derive efficient inversion recursions for sums of products of Toeplitz and Hankel matrices [9]. Merchant and Parks [10] reduced the solution of a system of equations with a Toeplitz-plus-Hankel coefficient matrix to the solution of a system of equations with a 2×2 block Toeplitz coefficient matrix, which is then solved by applying the block Levinson algorithm [11]. This procedure however, does not work for any strongly regular Toeplitz-plus-Hankel matrix, since it requires also the strong regularity of the corresponding Toeplitz-minus-Hankel matrix. Heinig, Jankowski, and Rost [12] also developed fast algorithms for the inversion of strongly regular Toeplitz-plus-Hankel matrices. Their recursive procedure is based on the concept of UV reduction and the solution of the so-called fundamental equations [13]. In [14, 15] Heinig and Rost represented Toeplitz-plus-Hankel inverses as sums of products of triangular Toeplitz and Hankel matrices, and in [16] Lev-Ari used the Schur reduction procedure to derive a fast algorithm for the triangular factorization of Hermitian Toeplitz-plus-Hankel matrices. Gohberg et al. [17] presented a fast algorithm for the solution of a system of linear equations with a strongly regular symmetric coefficient matrix which is the sum of a real Toeplitz matrix and a real Hankel matrix. Their derivation is based on the general approach developed in [17]. More recently, Yagle [18]

extended the split Levinson and Schur algorithms [19, 20] to strongly regular Toeplitz plus Hankel matrices, and Zarowski [21] used the algorithm of Heinig, Jankowski, and Rost [12] to induce a Schur-type recursion for Toeplitz-plus-Hankel matrices.

We remark that many earlier solutions to the factorization of Toeplitz-plus-Hankel matrices have been obtained by embedding the original matrix into a larger Toeplitz matrix. This approach, however, does not exhibit any nesting properties and so can only lead to fixed-order solutions. That is, if the matrix in question grows by one row and one column, then the factorization of the larger matrix has to be obtained by repeating the embedding procedure. In contrast, the algorithm presented in this paper allows for nested solutions: it can easily "update" the previous solution to reflect the change in the original matrix.

In this paper we give fast (and parallelizable) triangular factorization algorithms for a general class of strongly regular structured matrices which can be written as sums of quasi-Toeplitz and quasi-Hankel matrices. This class includes, among others, Toeplitz matrices, Hankel matrices, Toeplitz-plus-Hankel matrices, and the inverse of a Toeplitz matrix plus a Hankel matrix. Both the Hermitian and the non-Hermitian cases are considered. Our derivation is based on the observation that a quasi-Toeplitz matrix plus a quasi-Hankel matrix is structured, though in a more general sense than the "original" notion of displacement structure [22, 23], contrary to the statement made in [21]. We derive a general recursive algorithm (Algorithm 1 in Section 5.1) and then consider some important special cases such as Toeplitz plus Hankel matrices. We also exhibit the recursions in matrix form.

This paper is organized as follows. In Section 2 we review the definition of generating functions, quasi-Toeplitz, quasi-Hankel, and structured matrices, and introduce the family of matrices close to Toeplitz plus Hankel. In Section 3 we show how the Schur reduction procedure yields the triangular factorization of non-Hermitian matrices, and in Section 4 we derive fast triangular factorization algorithms for structured non-Hermitian matrices. These algorithms reduce in the Hermitian case to those already presented in [23]. In Section 5 the derived algorithms are applied to matrices close to Toeplitz plus Hankel, and in Section 6 we specialize these algorithms for quasi-Hankel and quasi-Toeplitz matrices.

2. PRELIMINARIES

In this section we review the definition of generating functions, quasi-Toeplitz matrices, quasi-Hankel matrices, and structured matrices, and intro-

duce the class of matrices close to Toeplitz plus Hankel. Our goal will be triangular factorization, which is a nested operation. Therefore we can assume, without loss of generality, that all matrices are extended to be semiinfinite.

Matrices T that are constant along the diagonals are called Toeplitz matrices. That is, the (i, j)th element is a function of $(i - j)$:

$$
T = \begin{bmatrix} c_0 & c_{-1} & c_{-2} & \cdots \\ c_1 & c_0 & c_{-1} & \\ c_2 & c_1 & c_0 & \\ \vdots & & & \ddots \end{bmatrix}.
$$

Similarly, matrices H that are constant along the antidiagonals are called Hankel matrices. That is, the (i, j)th element is a function of $(i + j)$:

$$
H = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots \\ h_1 & h_2 & h_3 & \\ h_2 & h_3 & h_4 & \\ \vdots & & & \ddots \end{bmatrix}.
$$

Note that a Hankel matrix H is always symmetric. H is Hermitian if, and only if, H is real, whereas T is Hermitian if, and only if, $c_{-i} = c_i^*$.

DEFINITION 1 (Generating function [23]).   The generating function of a semiinfinite matrix $R = [r_{ij}]_{i,j=0}^{\infty}$ is the bivariate function $R(z, w)$ defined by

$$
R(z, w) = \begin{bmatrix} 1 & z & z^2 & \cdots \end{bmatrix} R \begin{bmatrix} 1 & w & w^2 & \cdots \end{bmatrix}^*.
$$

It can be easily checked that the generating functions of Toeplitz and Hankel matrices are

$$
T(z, w) = \frac{c(z) + r^*(w)}{1 - zw^*} \quad \text{and} \quad H(z, w) = \frac{j[zh(z) - w^*h(w^*)]}{j(z - w^*)},
$$

$$
\tag{1}
$$

where $j = \sqrt{-1}$ and

$$h(z) = h_0 + zh_1 + z^2h_2 + \cdots,$$

$$c(z) = \frac{c_0}{2} + zc_1 + z^2c_2 + \cdots,$$

$$r^*(w) = \frac{c_0}{2} + w^*c_{-1} + w^{*2}c_{-2} + \cdots.$$

Notice that when H is Hermitian (or real), $h(z^*) = h^*(z)$.

DEFINTION 2 (Quasi-Toeplitz matrix [23]). A matrix QT is said to be quasi-Toeplitz if its generating function $QT(z, w)$ can be written in the form

$$QT(z, w) = a(z)T(z, w)b^*(w) \tag{2}$$

for some Toeplitz matrix T and functions $a(z)$ and $b(z)$. The matrix QT is Hermitian if $a(z) = b(z)$ and T is Hermitian.

DEFINITION 3 (Quasi-Hankel matrix [23]). A matrix QH is said to be quasi-Hankel if its generating function $QH(z, w)$ can be written in the form

$$QH(z, w) = \alpha(z)H(z, w)\beta^*(w) \tag{3}$$

for some Hankel matrix H and functions $\alpha(z)$ and $\beta(z)$. The matrix QH is Hermitian if $\alpha(z) = \beta(z)$ and H is real.

In this paper, we shall study matrices R whose generating functions can be written in the form

$$R(z, w) = QT(z, w) + QH(z, w).$$

That is, R is the sum of a quasi-Toeplitz matrix and a quasi-Hankel matrix. This clearly includes the sum of Toeplitz and Hankel matrices ($R = T + H$). It also includes matrices R of the form $R = T^{-1} + H$, since the inverse of a Toeplitz matrix is quasi-Toeplitz (see [23] and the references therein).

Using the expression (1) for $T(z, w)$ and $H(z, w)$, and the expressions (2)–(3), we obtain that $R(z, w)$ is of the form

$$R(z, w) = \frac{G(z)JB^*(w)}{j(1 - zw^*)(z - w^*)}, \tag{4}$$

where $G(z)$ and $B(z)$ are 1×8 row matrix functions

$$G(z) = \begin{bmatrix} a(z) & za(z) & \alpha(z) & z\alpha(z) & z^2\alpha(z)h(z) \end{bmatrix}$$

$$z\alpha(z)h(z) \quad a(z)c(z) \quad za(z)c(z) \end{bmatrix},$$

$$B(z) = \begin{bmatrix} b(z) & zb(z) & \beta(z) & z\beta(z) & z^2\beta(z)\tilde{h}(z) \end{bmatrix}$$

$$z\beta(z)\tilde{h}(z) \quad b(z)r(z) \quad zb(z)r(z) \end{bmatrix},$$

$$J = \begin{bmatrix} & & & & & & & -j \\ & & & & & & j & \\ & & & & & -j & & \\ & & & & j & & & \\ & & & -j & & & & \\ & & j & & & & & \\ & -j & & & & & & \\ j & & & & & & & \end{bmatrix},$$

and $\tilde{h}(z) = \sum_{i=0}^{\infty} h_i^* z^i$. Notice that $\tilde{h}(z) = h^*(z^*)$ is the same power series as $h(z)$ but with conjugate coefficients. If $a(z) = b(z)$, $\alpha(z) = \beta(z)$, and H is real, then $\tilde{h}(z) = h(z)$ and $G(z) = B(z)$, and thus R is Hermitian. In some special cases, the matrices $G(z)$ and $B(z)$ can have a lower column dimension. For example, if $a(z) = \alpha(z)$ and $b(z) = \beta(z)$, then

$$G(z) = \begin{bmatrix} a(z) & za(z) & a(z)[c(z) + z^2h(z)] & za(z)[c(z) + h(z)] \end{bmatrix},$$

$$B(z) = \begin{bmatrix} b(z) & zb(z) & b(z)[r(z) + z^2\tilde{h}(z)] & zb(z)[r(z) + \tilde{h}(z)] \end{bmatrix},$$

and

$$
J = \begin{bmatrix} & & & -j \\ & & j & \\ & -j & & \\ j & & & \end{bmatrix}.
$$

This special case includes Toeplitz-plus-Hankel matrices [$a(z) = b(z) = \alpha(z) = \beta(z) = 1$].

Equation (4) and its special cases belong to the more general family whose generating functions can be written as

$$
R(z, w) = \frac{G(z) J B^*(w)}{d(z, w)}, \tag{5}
$$

where J is any constant nonsingular matrix, $d(z, w)$ is the generating function of a constant (possibly singular) Hermitian matrix $\boldsymbol{d} = [d_{ij}]_{i,j=0}^{\infty}$, viz.,

$$
d(z, w) = \begin{bmatrix} 1 & z & z^2 & \cdots \end{bmatrix} \boldsymbol{d} \begin{bmatrix} 1 & w & w^2 & \cdots \end{bmatrix}^*,
$$

and $G(z)$ and $B(z)$ are $1 \times \rho$ row vector functions, where ρ is called the displacement rank of R. Equation (5) has a matrix-domain equivalent. If we define

$$
\nabla_d R = \sum_{i,j=0}^{\infty} d_{ij} Z^i R Z^{*j}, \tag{6}
$$

where Z is the lower triangular shift matrix with ones on the first subdiagonal, then

$$
\nabla_d R = G J B^* \quad \text{with} \quad \rho = \text{rank}(\nabla_d R),
$$

where G and B are matrices with ρ columns. Each column of G [respectively B] is formed by stacking the coefficients of the corresponding function in $G(z)$ [respectively $B(z)$], i.e.,

$$
G(z) = \begin{bmatrix} 1 & z & z^2 & \cdots \end{bmatrix} G \quad \text{and} \quad B(z) = \begin{bmatrix} 1 & z & z^2 & \cdots \end{bmatrix} B. \tag{7}
$$

The structure (5) allows a so-called fast Schur reduction procedure for triangular factorization. This is described in the next two sections.

3. THE SCHUR REDUCTION PROCEDURE

The Schur reduction is an $O(n^3)$ procedure that factors an $n \times n$ matrix R into the form

$$R = LDU,$$

where L is a lower triangular matrix with unit diagonal elements, U is an upper triangular matrix with unit diagonal elements, and D is a diagonal matrix. A necessary and sufficient condition for D to be nonsingular is that R be strongly regular (i.e., all leading principal submatrices of R are nonsingular). This condition will be assumed throughout this paper. For a strongly regular matrix $R = [r_{ij}]_{i,j=0}^{n-1}$ we have

$$R - \begin{bmatrix} r_{00} \\ r_{10} \\ r_{20} \\ \vdots \end{bmatrix} r_{00}^{-1} \begin{bmatrix} r_{00} & r_{01} & r_{02} & \cdots \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & & & & \\ \vdots & & \tilde{R}_1 & & \\ 0 & & & & \end{bmatrix}.$$

Let $d_0 = r_{00}$, and define

$$l_0 = \begin{bmatrix} r_{00} \\ r_{10} \\ r_{20} \\ \vdots \end{bmatrix} d_0^{-1}, \qquad u_0 = \begin{bmatrix} r_{00}^* \\ r_{01}^* \\ r_{02}^* \\ \vdots \end{bmatrix} d_0^{-*}, \quad \text{and} \quad R_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & & & & \\ \vdots & & \tilde{R}_1 & & \\ 0 & & & & \end{bmatrix};$$

then $R = d_0 l_0 u_0^* + R_1$. The matrix $\tilde{R}_1$ is called the Schur complement of R with respect to r_{00}. This suggests the following recursive procedure for the triangular factorization of R:

$$R_0 = R, \qquad R_{i+1} = R_i - d_i l_i u_i^*, \quad 0 \leqslant i \leqslant n - 1,$$

$$d_i = e_i^T R_i e_i, \qquad l_i = R_i e_i d_i^{-1}, \quad \text{and} \quad u_i^* = e_i^T R_i d_i^{-1}, \tag{8}$$

where $e_i = [0 \;\; \cdots \;\; 0 \;\; 1 \;\; 0 \;\; \cdots \;\; 0]^T$ has i leading zeros. Notice that the first i

rows and columns of R_i are zero, the first i elements of l_i and u_i are zero, and the $(i + 1)$th elements of l_i and u_i are both equal to 1. For an $n \times n$ matrix R, the matrices L, D, and U are given by

$$L = \begin{bmatrix} l_0 & l_1 & \cdots & l_{n-1} \end{bmatrix}, \qquad U = \begin{bmatrix} u_0 & u_1 & \cdots & u_{n-1} \end{bmatrix}^*,$$

$$D = \text{diag}\{d_0, d_1, \ldots, d_{n-1}\}.$$

The diagonal elements d_i are guaranteed to be nonzero by the strong regularity of R. For a matrix R that is not strongly regular the recursive procedure breaks down if either or both of the following situations arise:

$$d_i = 0 \quad \text{and} \quad R_i e_i \neq \mathbf{0},$$

$$d_i = 0 \quad \text{and} \quad e_i^T R_i \neq \mathbf{0}.$$

However, if $d_i = 0$, $R_i e_i = \mathbf{0}$, and $e_i^T R_i = \mathbf{0}$, then we can choose l_i and u_i arbitrarily. A convenient choice would be $l_i = u_i = e_i$, which is consistent with L and U having unit diagonal elements.

4. GENERATING FUNCTION APPROACH

The complexity of the Schur reduction procedure can be reduced to $O(\rho n^2)$ in the case of structured matrices as in (5). This is because the recursive procedure (8) reduces to a recursive update of the generator matrices G and B, which have ρn elements each, as compared to n^2 in R. We extend here the approach used in [23, 24] to the non-Hermitian case. Using the notation of generating functions (and assuming, without loss of generality, that R is extended to be semiinfinite as explained at the beginning of Section 2), it is easy to check that (8) can be written in the form

$$d_i = \tilde{R}_i(0, 0), \tag{9}$$

$$l_i(z) = z^i \tilde{R}_i(z, 0) d_i^{-1} \quad \text{and} \quad u_i(w) = w^{*i} \tilde{R}_i(0, w) d_i^{-1}, \tag{10}$$

$$zw^* \tilde{R}_{i+1}(z, w) = \tilde{R}_i(z, w) - \tilde{R}_i(z, 0) \tilde{R}_i^{-1}(0, 0) \tilde{R}_i(0, w), \tag{11}$$

where

$$l_i(z) = \begin{bmatrix} 1 & z & z^2 & \cdots \end{bmatrix} l_i,$$

$$u_i(w) = u_i^* \begin{bmatrix} 1 & w & w^2 & \cdots \end{bmatrix}^*,$$

$$\tilde{R}_i(z, w) = (zw^*)^{-i} R_i(z, w).$$

Notice that the triangular factors of the given finite matrix R are obtained by considering the first n coefficients of $l_i(z)$ and $u_i(z)$.

We shall now proceed by induction. Suppose

$$\tilde{R}_i(z, w) = \frac{G_i(z) J B_i^*(w)}{d(z, w)}$$

(this is certainly true for $i = 0$); then (11) yields

$$zw^* \tilde{R}_{i+1}(z, w) = \frac{G_i(z) \left(J - \dfrac{d(z, w)}{d(z, 0)\, d(0, w)} J M_i J \right) B_i^*(w)}{d(z, w)},$$

where

$$M_i = B_i^*(0) \tilde{R}_i^{-1}(0, 0) G_i(0).$$

Observe that M_i satisfies

$$M_i J M_i = d(0, 0) M_i. \tag{12}$$

If we can find matrices $\Theta_i(z)$ and $\Gamma_i(z)$ such that

$$\Theta_i(z) J \Gamma_i^*(w) = J - \frac{d(z, w)}{d(z, 0) d(0, w)} J M_i J, \tag{13}$$

then we can write

$$zw^* \tilde{R}_{i+1}(z, w) = \frac{G_i(z) \Theta_i(z) J \Gamma_i^*(w) B_i^*(w)}{d(z, w)}.$$

This shows that we can use the following recursions:

$$zG_{i+1}(z) = G_i(z)\Theta_i(z), \qquad G_0(z) = G(z),$$

$$zB_{i+1}(z) = B_i(z)\Gamma_i(z), \qquad B_0(z) = B(z) \tag{14}$$

for the update of $G_i(z)$ and $B_i(z)$. Moreover, d_i, $l_i(z)$, and $u_i(z)$ can be determined from $G_i(z)$ and $B_i(z)$ without the need to evaluate $\bar{R}_i(z, w)$ explicitly:

$$d_i = \lim_{z \to 0} \frac{G_i(z)JB_i^*(z)}{d(z, z)},$$

$$l_i(z) = z^i \frac{G_i(z)JB_i^*(0)}{d(z, 0)} d_i^{-1}, \quad \text{and} \quad u_i(w) = w^{*i} \frac{G_i(0)JB_i^*(w)}{d(0, w)} d_i^{-1}.$$

Since the matrices G_i and B_i have each $\rho(n - i)$ elements, then n steps of the recursions (14) require $O(\rho n^2)$ operations (additions and multiplications), which represents great savings in computation when $\rho \lessdot n$.

The existence of solutions $\Theta_i(z)$ and $\Gamma_i(z)$ to Equation (13) is guaranteed by the following theorem.

THEOREM 1 (Factorization). *Let R be a strongly regular structured matrix whose generating function is given by*

$$R(z, w) = \frac{G(z)JB^*(w)}{d(z, w)},$$

where J is a constant nonsingular matrix and $d(z, w)$ is a scalar bivariate Hermitian function of the form

$$d(z, w) = e(z)e^*(w) - f(z)f^*(w). \tag{15}$$

Then the triangular factorization of R can be carried out recursively using (14) with

$$\Theta_i(z) = \{I - \lambda(z)JM_i\}U_i \quad \text{and} \quad \Gamma_i(z) = \{I - \lambda(z)J^*M_i^*\}V_i$$

where

$$\lambda(z) = \frac{d(z, \tau)}{d(z, 0)d(0, \tau)},\tag{16}$$

U_i *and* V_i *are arbitrary constant matrices satisfying* $U_i J V_i^* = J$, *and* τ *is any scalar such that* $d(\tau, \tau) = 0$.

Proof. This theorem is an extension to the non-Hermitian case of Theorem 3 in [23] and of the discussion in [24]. Using the expressions for $\Theta_i(z)$ and $\Gamma_i(z)$ with (12), we have

$$\Theta_i(z)J\Gamma_i^*(w) = J - [\lambda(z) + \lambda^*(w) - \lambda(z)\lambda^*(w)d(0, 0)]JM_iJ.$$

Substituting into (13), we obtain

$$\frac{d(z, w)}{d(z, 0)d(0, w)} = \lambda(z) + \lambda^*(w) - d(0, 0)\lambda(z)\lambda^*(w).$$

This is the same as an equation obtained in [23, 24], where it is shown that a solution $\lambda(z)$ exists if, and only if, $d(z, w)$ is of the form (15). Moreover, $\lambda(z)$ is given by (16).   ∎

COROLLARY 1 (Generators of the Schur complement). G_i *and* B_i *are the generator matrices of* $\bar{R}_i$, *which is the Schur complement of* R *with respect to the leading* $i \times i$ *principal submatrix*.

5. FACTORIZATION OF QT + QH MATRICES

In this section we specialize the recursions of the previous theorem for strongly regular matrices R whose generating functions can be expressed in the form

$$R(z, w) = \frac{G(z)JB^*(w)}{j(1 - zw^*)(z - w^*)},\tag{17}$$

where G and B are matrices with ρ columns each, and J is a $\rho \times \rho$ matrix

with nonzero elements along the antidiagonal given by an alternating sequence of $-j$ and j:

$$
J = \begin{bmatrix} & & & -j \\ & & j & \\ & -j & & \\ \ddots & & & \end{bmatrix}.
$$

It is clear that J is Hermitian when ρ is even, and anti-Hermitian (i.e. $J^* = -J$) when ρ is odd. We shall denote by $\bar{J}$ the real matrix defined by $J = j\bar{J}$.

Next observe that $d(z, w)$ in (17) can be written as

$$
d(z, w) = \begin{bmatrix} 1 + z^2 & z \end{bmatrix} \begin{bmatrix} 0 & -j \\ j & 0 \end{bmatrix} \begin{bmatrix} 1 + w^{*2} \\ w^* \end{bmatrix}.
$$

This shows that $d(z, w)$ is in fact the generating function of a rank-2 Hermitian matrix **d**, which has one positive and one negative eigenvalue and thus is of the form (15). Hence, quasi-Toeplitz-plus-quasi-Hankel matrices are structured with respect to the following matrix-domain displacement (recall (6), and see [25, 26] for generalized definitions of displacement structure):

$$
\nabla_d R = ZR(I + Z^2) - (I + Z^2)RZ^*.
$$

We should remark, however, that the sum of two structured matrices, each one with a different $d(z, w)$, is not in general structured. That is, the resulting $d(z, w)$ need not in general have the form described by (15). Our derivation shows that for the special case of sums of QT and QH matrices, the corresponding QT + QH matrix is structured and hence the factorization procedure described in the previous section can be applied. Observe that this procedure has many degrees of freedom represented by the parameters U_i, V_i, and τ. In the sequel we shall make the simplest choices so as to simplify the final algorithms. We have

$$
\lambda(z) = j(1 - z\tau^*)\left(\frac{1}{\tau^*} - \frac{1}{z}\right),
$$

where τ is any solution of $(1 - \tau\tau^*)(\tau - \tau^*) = 0$. The simplest form of $\lambda(z)$ is obtained for $\tau = j$. In this case $\lambda(z) = -j(z + z^{-1})$. We are assuming that

R is strongly regular and thus $d_i = \tilde{R}_i(0,0) \neq 0$ for every i. But $d(0,0)$ in (17) is zero, which shows that

$$G_i(0)\tilde{J}B_i^*(0) = 0. \tag{18}$$

Therefore we need a way to compute $\tilde{R}_i(0,0)$.

LEMMA 1 (Diagonal factors). $\tilde{R}_i(0,0)$ *can be computed by either of the following two expressions*:

$$\tilde{R}_i(0,0) = G_i'(0)\tilde{J}B_i^*(0),$$

$$\tilde{R}_i(0,0) = -G_i(0)\tilde{J}B_i'^*(0),$$

where $G_i'(0)$ (respectively $B_i'(0)$) is the second row of the corresponding matrix G_i (respectively B_i).

Proof. The proof follows easily by noting that the column vectors l_i and u_i both have i leading zeros followed by 1. From (8) and (10) we get

$$1 = G_i'(0)\tilde{J}B_i^*(0)d_i^{-1} = -G_i(0)\tilde{J}B_i'^*(0)d_i^{-1}. \qquad \blacksquare$$

The choice of U_i and V_i is nonunique. We may choose $U_i = V_i = I$, which leads to an overall $O(4\rho n^2)$ computational complexity for each generator. An alternative choice would be to reduce the generators G_i and B_i to proper form. More specifically, since J is nonsingular and $U_iJV_i^* = J$ (which can be written also as $U_i\tilde{J}V_i^* = \tilde{J}$), the matrices U_i and V_i are nonsingular. Moreover, neither $G_i(0)$ nor $B_i(0)$ can be $\mathbf{0}$, because d_i would then be zero (by Lemma 1) and R would not be a strongly regular matrix. Hence it is always possible to find matrices U_i and V_i such that G_iU_i and B_iV_i are reduced to the forms

$$U_i\tilde{J}V_i^* = \tilde{J}, \tag{19}$$

$$G_i(0)U_i = \left|\tilde{R}_i(0,0)\right|^{1/2}[\,1 \quad 0 \quad \cdots \quad 0\,], \tag{20}$$

$$B_i(0)V_i = \left|\tilde{R}_i(0,0)\right|^{1/2}[\,1 \quad 0 \quad \cdots \quad 0\,]. \tag{21}$$

We can, for instance, use a sequence of elementary rotations to achieve (20)–(21). This also requires $O(4\rho n^2)$ operations per generator. We shall however give alternative (explicit) expressions for U_i and V_i for all $\rho \leqslant 8$.

Meanwhile, suppose we find matrices U_i and V_i that satisfy (19)–(21); then

$$\Theta_i(z) = U_i - (z + z^{-1})\tilde{J}B_i^*(0)\,\tilde{R}_i^{-1}(0,0)G_i(0)U_i$$

$$= U_i\left\{I - (z + z^{-1})U_i^{-1}\tilde{J}B_i^*(0)\,\tilde{R}_i^{-1}(0,0)G_i(0)U_i\right\}$$

$$= U_i\left\{I - (z + z^{-1})\tilde{J}V_i^*B_i^*(0)\,\tilde{R}_i^{-1}(0,0)G_i(0)U_i\right\}$$

$$= U_i S_\Theta(z), \tag{22}$$

where $S_\Theta(z)$ is the $\rho \times \rho$ matrix

$$S_\Theta(z) = I + \begin{bmatrix} & 0 & & & \\ & \vdots & & \mathbf{0} & \\ & 0 & & & \\ -\epsilon_i(z + z^{-1}) & 0 & \cdots & 0 \end{bmatrix}$$

and

$$\epsilon_i = \begin{cases} +\dfrac{\left|\tilde{R}_i(0,0)\right|}{\tilde{R}_i(0,0)} & \text{if } \rho \text{ is even,} \\[4mm] -\dfrac{\left|\tilde{R}_i(0,0)\right|}{\tilde{R}_i(0,0)} & \text{if } \rho \text{ is odd.} \end{cases} \tag{23}$$

Similarly we can show that $\Gamma_i(z)$ reduces to

$$\Gamma_i(z) = V_i S_\Gamma(z) \tag{24}$$

where

$$
S_\Gamma(z) = I + \begin{bmatrix} 0 \\ \vdots & & \mathbf{0} \\ 0 \\ -\varepsilon_i(z + z^{-1}) & 0 & \cdots & 0 \end{bmatrix} \quad \text{and} \quad \varepsilon_i = \frac{|\tilde{R}_i(0,0)|}{\tilde{R}_i^*(0,0)}.
$$

5.1. Matrix Forms

It is easy to verify that the generator recursions (14) [along with (22) and (24)] have the following matrix representations:

$$
\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = G_i U_i - \epsilon_i (Z + Z^*) G_i U_i \begin{bmatrix} 0_{p-1} \\ 1 \end{bmatrix},
$$

$$
\begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} = B_i V_i - \varepsilon_i (Z + Z^*) B_i V_i \begin{bmatrix} 0_{p-1} \\ 1 \end{bmatrix},
$$

where $G_i(z)$ and G_i [similarly for $B_i(z)$ and B_i] are related as in (7). This shows that G_{i+1} is obtained as follows (a similar argument holds for B_{i+1}):

(a) We multiply G_i by U_i and keep the last $p - 1$ columns.

(b) The last column of $G_i U_i$ is shifted up (introducing a zero at the bottom) and shifted down (introducing a zero at the top). The two shifted versions are added together and multiplied by $-\epsilon_i$. The result is then added to the first column of $G_i U_i$. This gives the first column of G_{i+1}.

Moreover, the nonzero parts of the (semiinfinite) triangular factors l_i and u_i (denoted by $\tilde{l}_i$ and $\tilde{u}_i$ respectively),

$$
l_i = \begin{bmatrix} \mathbf{0}_{i \times 1} \\ \tilde{l}_i \end{bmatrix} \quad \text{and} \quad u_i = \begin{bmatrix} \mathbf{0}_{i \times 1} \\ \tilde{u}_i \end{bmatrix},
$$

are given by

$$
\tilde{l}_i(z) = z^{-1} G_i(z) \tilde{J} B_i^*(0) d_i^{-1},
$$

$$
\tilde{u}_i(w) = -w^{-*} G_i(0) \tilde{J} B_i^*(w) d_i^{-1},
$$

which can be rewritten in matrix form as well. Consider the expression for $\tilde{l}_i(z)$ [the same argument holds for $\bar{u}_i(w)$]. We write

$$\tilde{l}_i = Z^* G_i U_i \tilde{J} V_i^* B_i^*(0) \tilde{R}_i^{-1}(0,0)$$

$$= \epsilon_i \left| \tilde{R}_i(0,0) \right|^{-1/2} Z^* g_i,$$

where g_i denotes the last column of $G_i U_i$. Recall that $\tilde{l}_i$ is a column vector starting with 1. Hence the second element of g_i must be equal to $(\epsilon_i | \tilde{R}_i(0,0) |^{-1/2})^{-1}$. This follows also from Lemma 1. Similarly, we can show that

$$\tilde{u}_i = \varepsilon_i \left| \tilde{R}_i(0,0) \right|^{-1/2} Z^* b_i,$$

where b_i is the last column of $B_i V_i$. Moreover, the second element of b_i must be equal to $(\varepsilon_i | \tilde{R}_i(0,0) |^{-1/2})^{-1}$.

In summary, we have the following algorithm.

ALGORITHM 1 (Factorization of QT + QH).   The triangular factorization of a matrix R with generating function $R(z, w)$ as in (17) can be carried out recursively as follows:

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = G_i U_i - \epsilon_i (Z + Z^*) G_i U_i \begin{bmatrix} 0_{p-1} \\ 1 \end{bmatrix}, \qquad G_0 = G,$$

$$\begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} = B_i V_i - \varepsilon_i (Z + Z^*) B_i V_i \begin{bmatrix} 0_{p-1} \\ 1 \end{bmatrix}, \qquad B_0 = B.$$

Let g_i and b_i denote the last columns of $G_i U_i$ and $B_i V_i$ respectively, and let g_{i2} and b_{i2} denote the second elements of g_i and b_i respectively. Then

$$\tilde{l}_i = Z^* \frac{g_i}{g_{i2}} \quad \text{and} \quad \bar{u}_i = Z^* \frac{b_i}{b_{i2}}.$$

The diagonal factor d_i [or $\tilde{R}_i(0,0)$] is computed as in Lemma 1, and ϵ_i and ε_i as defined above.

This leaves only the problem of finding U_i and V_i. As we mentioned earlier, there are many possible choices. We shall give explicit expressions for

two typical examples ($\rho = 4, 5$) and the same approach holds for other cases that we list in Appendix A. Meanwhile, it is worth noting that the recursions in the algorithm get simplified in the following cases:

(a) ρ is even and $G(z) = B(z)$: in this case R is Hermitian, and it follows from (8) that $l_i = u_i$ and $d_i^* = d_i$ (i.e., d_i is real for all i).

(b) ρ is odd and $G(z) = B(z)$: in this case R is anti-Hermitian, and it follows from (8) that $l_i = u_i$ and $d_i^* = -d_i$ (i.e., d_i is imaginary for all i).

EXAMPLE 1 (Displacement rank 4). This example includes the case of Toeplitz-plus-Hankel matrices. Define the quantities

$$\begin{bmatrix} \gamma_i^* & -\delta_i^* & \eta_i^* & -\kappa_i^* \end{bmatrix} = \left| \tilde{R}_i(0,0) \right|^{-1/2} G_i(0),$$

$$\begin{bmatrix} \mu_i^* & -\sigma_i^* & \chi_i^* & -\psi_i^* \end{bmatrix} = \left| \tilde{R}_i(0,0) \right|^{-1/2} B_i(0),$$

$$U_i = \begin{bmatrix} \gamma_i^{-*} & \delta_i^* & \delta_i^* - \dfrac{\eta_i^*}{\mu_i \gamma_i^*} & \psi_i \\ 0 & \gamma_i^* & \gamma_i^* & \chi_i \\ 0 & 0 & \mu_i^{-1} & \sigma_i \\ 0 & 0 & 0 & \mu_i \end{bmatrix}, \quad \text{and}$$

$$V_i = \begin{bmatrix} \mu_i^{-*} & \sigma_i^* & \sigma_i^* - \dfrac{\chi_i^*}{\gamma_i \mu_i^*} & \kappa_i \\ 0 & \mu_i^* & \mu_i^* & \eta_i \\ 0 & 0 & \gamma_i^{-1} & \delta_i \\ 0 & 0 & 0 & \gamma_i \end{bmatrix}.$$

From (18), it follows that

$$-\kappa_i^* \mu_i + \eta_i^* \sigma_i - \delta_i^* \chi_i + \gamma_i^* \psi_i = 0. \tag{25}$$

It is easy to check with the help of (25) that (19)–(21) are satisfied. Multiplying G_i by U_i (similarly for $B_i V_i$) requires $O((\rho^2/2)(n - i))$ operations, which leads to an overall $O((\rho^2/2)n^2)$ complexity. In this example $\rho = 4$; hence we need $O(8n^2)$ operations, which is of the same order as $O(12n^2)$ obtained by choosing $U_i = V_i = I$, or by using a sequence of elementary rotations.

For the case of Toeplitz-plus-Hankel matrices, we start the recursions with

$$G(z) = \begin{bmatrix} 1 & z & c(z) + z^2 h(z) & z[c(z) + h(z)] \end{bmatrix},$$

$$B(z) = \begin{bmatrix} 1 & z & r(z) + z^2 \tilde{h}(z) & z[r(z) + \tilde{h}(z)] \end{bmatrix},$$

or equivalently, we start the matrix recursions of Algorithm 1 with the $2n \times 4$ generators G and B given by

$$G = \begin{bmatrix} 1 & 0 & \dfrac{c_0}{2} & 0 \\[2ex] 0 & 1 & c_1 & \dfrac{c_0}{2} + h_0 \\[2ex] 0 & 0 & c_2 + h_0 & c_1 + h_1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & c_{n-1} + h_{n-3} & c_{n-2} + h_{n-2} \\ 0 & 0 & h_{n-2} & c_{n-1} + h_{n-1} \\ 0 & 0 & h_{n-1} & h_n \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & h_{2(n-2)} & h_{2(n-1)} \end{bmatrix}$$

and

$$B = \begin{bmatrix} 1 & 0 & \dfrac{c_0}{2} & 0 \\[2ex] 0 & 1 & c^*_{-1} & \dfrac{c_0}{2} + h^*_0 \\[2ex] 0 & 0 & c^*_{-2} + h^*_0 & c^*_{-1} + h^*_1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & c^*_{-n+1} + h^*_{n-3} & c^*_{-n+2} + h^*_{n-2} \\ 0 & 0 & h^*_{n-2} & c^*_{-n+1} + h^*_{n-1} \\ 0 & 0 & h^*_{n-1} & h^*_n \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & h^*_{2(n-2)} & h^*_{2(n-1)} \end{bmatrix}.$$

EXAMPLE 2 (Displacement rank 5).   Define the quantities

$$\begin{bmatrix} \gamma_i^* & -\delta_i^* & \eta_i^* & -\kappa_i^* & \xi_i^* \end{bmatrix} = \left| \tilde{R}_i(0,0) \right|^{-1/2} G_i(0),$$

$$\begin{bmatrix} \mu_i^* & -\sigma_i^* & \chi_i^* & -\psi_i^* & \varphi_i^* \end{bmatrix} = \left| \tilde{R}_i(0,0) \right|^{-1/2} B_i(0),$$

$$U_i = \begin{bmatrix} \gamma_i^{-*} & \gamma_i^{-*}\delta_i^* & -\gamma_i^{-*}\eta_i^* & \gamma_i^{-*}\kappa_i^* & \varphi_i \\ 0 & 1 & 0 & 0 & \psi_i \\ 0 & 0 & 1 & 0 & \chi_i \\ 0 & 0 & 0 & 1 & \sigma_i \\ 0 & 0 & 0 & 0 & \mu_i \end{bmatrix},$$

$$V_i = \begin{bmatrix} \mu_i^{-*} & \mu_i^{-*}\sigma_i^* & -\mu_i^{-*}\chi_i^* & \mu_i^{-*}\psi_i^* & \xi_i \\ 0 & 1 & 0 & 0 & \kappa_i \\ 0 & 0 & 1 & 0 & \eta_i \\ 0 & 0 & 0 & 1 & \delta_i \\ 0 & 0 & 0 & 0 & \gamma_i \end{bmatrix}.$$

From (18), it follows that

$$-\xi_i^*\mu_i + \kappa_i^*\sigma_i - \eta_i^*\chi_i + \delta_i^*\psi_i - \gamma_i^*\varphi_i = 0. \tag{26}$$

It is easy to check with the help of (26) that (19)–(21) are satisfied. Observe that U_i and V_i are sparse matrices. This leads to an overall $O(4\rho n^2)$ computational complexity.

6.   FACTORIZATION OF QT AND QH MATRICES

Algorithms for the triangular factorization of quasi-Toeplitz matrices (QT) and quasi-Hankel matrices (QH) follow readily as special cases of the general recursive procedure described by Theorem 1.

6.1.   Quasi-Hankel Matrices
According to Equations (1) and (3) the generating function of a quasi-Hankel matrix can be expressed in the form

$$QH(z, w) = \frac{G(z)JB^*(w)}{j(z - w^*)},$$

where

$$J = \begin{bmatrix} 0 & -j \\ j & 0 \end{bmatrix},$$

$$G(z) = \begin{bmatrix} \alpha(z) & z\alpha(z)h(z) \end{bmatrix}, \quad \text{and} \quad B(z) = \begin{bmatrix} \beta(z) & z\beta(z)\bar{h}(z) \end{bmatrix}.$$

Notice that $d(z,w) = j(z - w^*)$ can be written as

$$d(z,w) = \begin{bmatrix} 1 & z \end{bmatrix} J \begin{bmatrix} 1 \\ w^* \end{bmatrix},$$

which shows that $d(z, w)$ satisfies (15), since J has one positive and one negative eigenvalue. Following steps similar to those at the beginning of Section 5, we get

$$\lambda(z) = j\left(\frac{1}{\tau^*} - \frac{1}{z} \right),$$

which simplifies to $\lambda(z) = -jz^{-1}$ on setting $\tau = \infty$ [note that any choice of τ with Im $\tau = 0$ satisfies $d(\tau, \tau) = 0$]. Now define

$$QH_i(z,w) = \frac{G_i(z) J B_i^*(w)}{j(z - w^*)},$$

$$\begin{bmatrix} \gamma_i^* & -\delta_i^* \end{bmatrix} = |QH_i(0,0)|^{-1/2} G_i(0),$$

$$\begin{bmatrix} \mu_i^* & -\sigma_i^* \end{bmatrix} = |QH_i(0,0)|^{-1/2} B_i(0),$$

$$U_i = \begin{bmatrix} \gamma_i^{-*} & \sigma_i \\ 0 & \mu_i \end{bmatrix}, \quad \text{and} \quad V_i = \begin{bmatrix} \mu_i^{-*} & \delta_i \\ 0 & \gamma_i \end{bmatrix}.$$

Then it is easy to check that

$$U_i J V_i^* = J,$$

$$G_i(0) U_i = |QH_i(0,0)|^{1/2} \begin{bmatrix} 1 & 0 \end{bmatrix},$$

$$B_i(0) V_i = |QH_i(0,0)|^{1/2} \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

and thus we get the following result (which can be written in matrix form as well).

ALGORITHM 2 (Factorization of QH matrices).   The triangular factorization of a strongly regular quasi-Hankel matrix QH can be carried out by the following recursive procedure:

$$zG_{i+1}(z) = G_i(z)\begin{bmatrix} \gamma_i^{-*} & \sigma_i \\ 0 & \mu_i \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -\epsilon_i z^{-1} & 1 \end{bmatrix},$$

$$zB_{i+1}(z) = B_i(z)\begin{bmatrix} \mu_i^{-*} & \delta_i \\ 0 & \gamma_i \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -\epsilon_i^* z^{-1} & 1 \end{bmatrix},$$

$$d_i = G_i'(0)\tilde{J}B_i^*(0), \qquad \epsilon_i = \frac{|d_i|}{d_i}$$

$$l_i(z) = z^{i-1}G_i(z)\tilde{J}B_i^*(0)d_i^{-1}, \quad \text{and}$$

$$u_i(w) = -(w^*)^{i-1}G_i(0)\tilde{J}B_i^*(w)d_i^{-1},$$

where $J = j\tilde{J}$. Note that when QH is Hermitian we get $G_i(z) = B_i(z)$, $U_i = V_i$, and the previous recursions reduce to those given in [23].

6.2.   Quasi-Toeplitz Matrices
    The generating function of a quasi-Toeplitz matrix can be expressed in the form

$$QT(z, w) = \frac{G(z)JB^*(w)}{1 - zw^*},$$

where $G(z)$ and $B(z)$ are 1×2 row vector functions, and

$$J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Notice that $d(z, w) = 1 - zw^*$ can be written as

$$d(z, w) = \begin{bmatrix} 1 & z \end{bmatrix} J \begin{bmatrix} 1 \\ w^* \end{bmatrix},$$

which shows that $d(z, w)$ satisfies (15), since J has one positive and one negative eigenvalue. Following steps similar to those at the beginning of Section 5, we get

$$\lambda(z) = 1 - z\tau^*,$$

which simplifies to $\lambda(z) = 1 - z$ on setting $\tau = 1$ [note that any choice of τ with $|\tau| = 1$ satisfies $d(\tau, \tau) = 0$]. Now define

$$\mathrm{QT}_i(z, w) = \frac{G_i(z) J B_i^*(w)}{1 - zw^*},$$

$$G_i(z) = \begin{bmatrix} s_i(z) & v_i(z) \end{bmatrix} = \begin{bmatrix} 1 & z & z^2 & \cdots \end{bmatrix} \begin{bmatrix} s_{ii} & v_{ii} \\ s_{i+1,i} & v_{i+1,i} \\ \vdots & \vdots \end{bmatrix},$$

$$B_i(z) = \begin{bmatrix} x_i(z) & y_i(z) \end{bmatrix} = \begin{bmatrix} 1 & z & z^2 & \cdots \end{bmatrix} \begin{bmatrix} x_{ii} & y_{ii} \\ x_{i+1,i} & y_{i+1,i} \\ \vdots & \vdots \end{bmatrix},$$

$$k_i = v_{ii}/s_{ii} \quad \text{and} \quad \xi_i = y_{ii}/x_{ii},$$

$$U_i = \frac{1}{1 - k_i \xi_i^*} \begin{bmatrix} 1 & -k_i \\ -\xi_i^* & 1 \end{bmatrix} \quad \text{and} \quad V_i = \begin{bmatrix} 1 & -\xi_i \\ -k_i^* & 1 \end{bmatrix}.$$

Then it is easy to check that

$$U_i J V_i^* = J,$$

$$G_i(0) U_i = s_{ii} \begin{bmatrix} 1 & 0 \end{bmatrix},$$

$$B_i(0) V_i = x_{ii}(1 - k_i^* \xi_i) \begin{bmatrix} 1 & 0 \end{bmatrix},$$

$$s_{ii} x_{ii}^* (1 - k_i \xi_i^*) = \mathrm{QT}_i(0, 0),$$

which yields the following result.

ALGORITHM 3 (Factorization of QT matrices).   The triangular factoriza-
tion of a strongly regular quasi-Toeplitz matrix QT can be carried out by the
following recursive procedure:

$$zG_{i+1}(z) = \frac{1}{1 - k_i \xi_i^*} G_i(z) \begin{bmatrix} 1 & -k_i \\ -\xi_i^* & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix},$$

$$zB_{i+1}(z) = B_i(z) \begin{bmatrix} 1 & -\xi_i \\ -k_i^* & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix},$$

$$d_i = G_i(0) J B_i^*(0),$$

$$l_i(z) = z^i G_i(z) J B_i^*(0) d_i^{-1} \quad \text{and} \quad u_i(w) = (w^*)^i G_i(0) J B_i^*(w) d_i^{-1},$$

$$k_i = v_{ii}/s_{ii} \quad \text{and} \quad \xi_i = y_{ii}/x_{ii}.$$

Note that when QT is Hermitian positive definite we get $G_i(z) = B_i(z)$,
$k_i = \xi_i$, and the previous recursions reduce to

$$zG_{i+1}(z) = G_i(z) \frac{1}{\sqrt{1 - |k_i|^2}} \begin{bmatrix} 1 & -k_i \\ -k_i^* & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix}, \tag{27}$$

$$k_i = v_{ii}/s_{ii},$$

which is the linearized form of the conventional Schur recursion [23, 27, 28].
If we define $\bar{v}_i(z) = z^i v_i(z)$, $\bar{s}_i(z) = z^i s_i(z)$, and $f_i(z) = \bar{v}_i(z)/\bar{s}_i(z)$, then
(27) reduces to the classical Schur recursion [29] for functions that are
analytic and bounded by unity in the unit disc,

$$f_{i+1}(z) = \frac{1}{z} \frac{f_i(z) - k_i}{1 - k_i^* f_i(z)} \quad \text{with} \quad k_i = f_i(0).$$

The positive definiteness of QT guarantees $|k_i| < 1$.
    The recursions for the non-Hermitian quasi-Toeplitz case are the same
expressions derived in [30], though from a different point of view. We remark
that the recursions for $G_i(z)$ and $B_i(z)$ differ by a scaling factor. One can,
however, choose different matrices U_i and V_i so that the resulting recursions
for the generators G_i and B_i become similar.

7.   CONCLUSION

In this paper we have presented fast triangular factorization algorithms $[O(n^2)]$ for matrices close to Toeplitz plus Hankel, and in particular for quasi-Hankel, quasi-Toeplitz, and Toeplitz-plus-Hankel matrices. Both Hermitian and non-Hermitian matrices were considered. The arguments are based on an extension to the non-Hermitian case of the generating-function approach discussed in [23]. The results include many special cases studied separately in the literature. We remark that we can, as well, compute the triangular factorization of the inverse of an $n \times n$ close to Toeplitz-plus-Hankel matrix R. For this purpose, we define the extended matrix

$$\begin{bmatrix} -R & I \\ I & 0 \end{bmatrix}$$

and observe that its Schur complement with respect to the $(1, 1)$ block entry is R^{-1}. We can exploit the structure of this extended matrix in the derivation of a fast factorization algorithm for R^{-1}; similar ideas can be used for solving linear equations with R as the coefficient matrix, finding orthogonal factorizations, etc. See [31] for details.

APPENDIX A

We list here possible choices of U_i and V_i for other displacement ranks than the cases $\rho = 4$ and $\rho = 5$ treated in Section 5.

Displacement rank 2:

$$\begin{bmatrix} \gamma_i^* & -\delta_i^* \end{bmatrix} = \left| \tilde{R}_i(0,0) \right|^{-1/2} G_i(0),$$

$$\begin{bmatrix} \mu_i^* & -\sigma_i^* \end{bmatrix} = \left| \tilde{R}_i(0,0) \right|^{-1/2} B_i(0),$$

$$U_i = \begin{bmatrix} \gamma_i^{-*} & \sigma_i \\ 0 & \mu_i \end{bmatrix}, \quad \text{and} \quad V_i = \begin{bmatrix} \mu_i^{-*} & \delta_i \\ 0 & \gamma_i \end{bmatrix}.$$

Displacement rank 3:

$$\left[\begin{matrix} \gamma_i^* & -\delta_i^* & \eta_i^* \end{matrix}\right] = \left|\tilde{R}_i(0,0)\right|^{-1/2} G_i(0),$$

$$\left[\begin{matrix} \mu_i^* & -\sigma_i^* & \chi_i^* \end{matrix}\right] = \left|\tilde{R}_i(0,0)\right|^{-1/2} B_i(0),$$

$$U_i = \begin{bmatrix} \gamma_i^{-*} & \gamma_i^{-*}\delta_i^* & \chi_i \\ 0 & 1 & \sigma_i \\ 0 & 0 & \mu_i \end{bmatrix}, \quad \text{and} \quad V_i = \begin{bmatrix} \mu_i^{-*} & \mu_i^{-*}\sigma_i^* & \eta_i \\ 0 & 1 & \delta_i \\ 0 & 0 & \gamma_i \end{bmatrix}.$$

Displacement rank 6:

$$\left[\begin{matrix} \gamma_i^* & -\delta_i^* & \eta_i^* & -\kappa_i^* & \xi_i^* & -\vartheta_i^* \end{matrix}\right] = \left|\tilde{R}_i(0,0)\right|^{-1/2} G_i(0),$$

$$\left[\begin{matrix} \mu_i^* & -\sigma_i^* & \chi_i^* & -\psi_i^* & \varphi_i^* & -\theta_i^* \end{matrix}\right] = \left|\tilde{R}_i(0,0)\right|^{-1/2} B_i(0),$$

$$U_i = \begin{bmatrix} \gamma_i^{-*} & \delta_i^* & \delta_i^* - \dfrac{\eta_i^*}{\mu_i\gamma_i^*} & \kappa_i^* - \dfrac{\eta_i^*}{\mu_i\gamma_i^*} & \delta_i^* + \kappa_i^* - \dfrac{\xi_i^* + \eta_i^*}{\mu_i\gamma_i^*} & \theta_i \\ 0 & \gamma_i^* & \gamma_i^* & 0 & \gamma_i^* & \varphi_i \\ 0 & 0 & \mu_i^{-1} & \mu_i^{-1} & \mu_i^{-1} & \psi_i \\ 0 & 0 & 0 & \gamma_i^* & \gamma_i^* & \chi_i \\ 0 & 0 & 0 & 0 & \mu_i^{-1} & \sigma_i \\ 0 & 0 & 0 & 0 & 0 & \mu_i \end{bmatrix},$$

$$V_i = \begin{bmatrix} \mu_i^{-*} & \sigma_i^* & \sigma_i^* - \dfrac{\chi_i^*}{\gamma_i\mu_i^*} & \psi_i^* - \dfrac{\chi_i^*}{\gamma_i\mu_i^*} & \sigma_i^* + \psi_i^* - \dfrac{\varphi_i^* + \chi_i^*}{\gamma_i\mu_i^*} & \vartheta_i \\ 0 & \mu_i^* & \mu_i^* & 0 & \mu_i^* & \xi_i \\ 0 & 0 & \gamma_i^{-1} & \gamma_i^{-1} & \gamma_i^{-1} & \kappa_i \\ 0 & 0 & 0 & \mu_i^* & \mu_i^* & \eta_i \\ 0 & 0 & 0 & 0 & \gamma_i^{-1} & \delta_i \\ 0 & 0 & 0 & 0 & 0 & \gamma_i \end{bmatrix}.$$

Displacement rank 7:

$$\begin{bmatrix} \gamma_i^* & -\delta_i^* & \eta_i^* & -\kappa_i^* & \xi_i^* & -\vartheta_i^* & \nu_i^* \end{bmatrix} = \left| \tilde{R}_i(0,0) \right|^{-1/2} G_i(0),$$

$$\begin{bmatrix} \mu_i^* & -\sigma_i^* & \chi_i^* & -\psi_i^* & \varphi_i^* & -\theta_i^* & s_i^* \end{bmatrix} = \left| \tilde{R}_i(0,0) \right|^{-1/2} B_i(0),$$

$$U_i = \begin{bmatrix} \gamma_i^{-*} & \gamma_i^{-*}\delta_i^* & -\gamma_i^{-*}\eta_i^* & \gamma_i^{-*}\kappa_i^* & -\gamma_i^{-*}\xi_i^* & \gamma_i^{-*}\vartheta_i^* & s_i \\ 0 & 1 & 0 & 0 & 0 & 0 & \theta_i \\ 0 & 0 & 1 & 0 & 0 & 0 & \varphi_i \\ 0 & 0 & 0 & 1 & 0 & 0 & \psi_i \\ 0 & 0 & 0 & 0 & 1 & 0 & \chi_i \\ 0 & 0 & 0 & 0 & 0 & 1 & \sigma_i \\ 0 & 0 & 0 & 0 & 0 & 0 & \mu_i \end{bmatrix},$$

$$V_i = \begin{bmatrix} \mu_i^{-*} & \mu_i^{-*}\sigma_i^* & -\mu_i^{-*}\chi_i^* & \mu_i^{-*}\psi_i^* & -\mu_i^{-*}\varphi_i^* & \mu_i^{-*}\theta_i^* & \nu_i \\ 0 & 1 & 0 & 0 & 0 & 0 & \vartheta_i \\ 0 & 0 & 1 & 0 & 0 & 0 & \xi_i \\ 0 & 0 & 0 & 1 & 0 & 0 & \kappa_i \\ 0 & 0 & 0 & 0 & 1 & 0 & \eta_i \\ 0 & 0 & 0 & 0 & 0 & 1 & \delta_i \\ 0 & 0 & 0 & 0 & 0 & 0 & \gamma_i \end{bmatrix}.$$

Displacement rank 8:

$$\begin{bmatrix} \gamma_i^* & -\delta_i^* & \eta_i^* & -\kappa_i^* & \xi_i^* & -\vartheta_i^* & \nu_i^* & -\varrho_i^* \end{bmatrix}$$

$$= \left| \tilde{R}_i(0,0) \right|^{-1/2} G_i(0),$$

$$\begin{bmatrix} \mu_i^* & -\sigma_i^* & \chi_i^* & -\psi_i^* & \varphi_i^* & -\theta_i^* & s_i^* & -\varpi_i^* \end{bmatrix}$$

$$= \left| \tilde{R}_i(0,0) \right|^{-1/2} B_i(0),$$

$$U_i = \begin{bmatrix} \gamma_i^{-*} & \delta_i^* & \delta_i^* - \dfrac{\eta_i^*}{\mu_i \gamma_i^*} & \kappa_i^* - \eta_i^* & u_{i1} & u_{i2} & u_{i3} & \varpi_i \\ 0 & \gamma_i^* & \gamma_i^* & 0 & \gamma_i^* & \gamma_i^* & \gamma_i^* & s_i \\ 0 & 0 & \mu_i^{-1} & \gamma_i^* & \gamma_i^* & \mu_i^{-1} & \gamma_i^* & \theta_i \\ 0 & 0 & 0 & \gamma_i^* & \gamma_i^* & 0 & \gamma_i^* & \varphi_i \\ 0 & 0 & 0 & 0 & \mu_i^{-1} & \gamma_i^* & \gamma_i^* & \psi_i \\ 0 & 0 & 0 & 0 & 0 & \gamma_i^* & \gamma_i^* & \chi_i \\ 0 & 0 & 0 & 0 & 0 & 0 & \mu_i^{-1} & \sigma_i \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_i \end{bmatrix},$$

$$V_i = \begin{bmatrix} \mu_i^{-*} & \sigma_i^* & \sigma_i^* - \dfrac{\chi_i^*}{\gamma_i \mu_i^*} & \psi_i^* - \xi_i^* & v_{i1} & v_{i2} & v_{i3} & \varrho_i \\ 0 & \mu_i^* & \mu_i^* & 0 & \mu_i^* & \mu_i^* & \mu_i^* & \nu_i \\ 0 & 0 & \gamma_i^{-1} & \mu_i^* & \mu_i^* & \gamma_i^{-1} & \mu_i^* & \vartheta_i \\ 0 & 0 & 0 & \mu_i^* & \mu_i^* & 0 & \mu_i^* & \xi_i \\ 0 & 0 & 0 & 0 & \gamma_i^{-1} & \mu_i^* & \mu_i^* & \kappa_i \\ 0 & 0 & 0 & 0 & 0 & \mu_i^* & \mu_i^* & \eta_i \\ 0 & 0 & 0 & 0 & 0 & 0 & \gamma_i^{-1} & \delta_i \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma_i \end{bmatrix},$$

where

$$u_{i1} = \delta_i^* + \kappa_i^* - \eta_i^* - \frac{\xi_i^*}{\mu_i \gamma_i^*}, \qquad u_{i2} = \vartheta_i^* + \delta_i^* - \xi_i^* - \frac{\eta_i^*}{\mu_i \gamma_i^*},$$

$$u_{i3} = \vartheta_i^* + \delta_i^* - \xi_i^* + \kappa_i^* - \eta_i^* - \frac{\nu_i^*}{\mu_i \gamma_i^*},$$

$$v_{i1} = \sigma_i^* + \psi_i^* - \chi_i^* - \frac{\varphi_i^*}{\gamma_i \mu_i^*}, \qquad v_{i2} = \theta_i^* + \sigma_i^* - \varphi_i^* - \frac{\chi_i^*}{\gamma_i \mu_i^*},$$

$$v_{i3} = \theta_i^* + \sigma_i^* - \varphi_i^* + \psi_i^* - \chi_i^* - \frac{s_i^*}{\gamma_i \mu_i^*}.$$

REFERENCES

1 I. M. Gelfand and B. M. Levitan, On the determination of a differential equation from its spectral function, *Amer. Math. Soc. Transl.* 1:253–304 (1955).

2   A. Bruckstein and T. Kailath, Inverse scattering for discrete transmission-line models, *SIAM Rev.* 29(3):359–389 (Sept. 1987).

3   B. Friedlander and M. Morf, Least-squares algorithms for adaptive linear-phase filtering, *IEEE Trans. Acoust. Speech Signal Process.* 30(3):381–389 (June 1982).

4   S. M. Kay and S. L. Marple, Spectrum analysis—a modern perspective, *Proc. IEEE* 69(11):1380–1419 (Nov. 1981).

5   S. L. Marple, Jr. A new autoregressive spectrum analysis algorithm, *IEEE Trans. Acoust. Speech Signal Process.* 28:441–454 (Aug. 1980).

6   JP. Durge, A. A. Beex, and L. L. Scharf, Generating covariance sequences and the calculation of quantization and rounding error variances in digital filters, *IEEE Trans. Acoust. Speech Signal Process.* 28(1):102–104 (Feb. 1980).

7   B. Friedlander, T. Kailath, M. Morf, and L. Ljung, Extended Levinson and Chandrasekhar equations for general discrete-time linear estimation problems, *IEEE Trans. Automat. Control* 23(4):653–659 (Aug. 1978).

8   M. Morf, Fast Algorithms for Multivariable Systems, Ph.D. Thesis, Stanford Univ., Stanford, Calif., 1974.

9   B. Friedlander and M. Morf, Efficient inversion formulas for sums of products of Toeplitz and Hankel matrices, in *Proceedings 18th Annual Allerton Conference on Communication, Control and Computing*, Oct. 1980, pp. 574–583.

10   G. A. Merchant and T. W. Parks, Efficient solution of a Toeplitz-plus-Hankel coefficient system of equations, *IEEE Trans. Acoust. Speech Signal Process.* 30(1):40–44 (Feb. 1982).

11   H. Akaike, Block Toeplitz matrix inversion, *SIAM J. Appl. Math.* 24:234–241 (Mar. 1973).

12   G. Heinig, P. Jankowski, and K. Rost, Fast inversion algorithms of Toeplitz-plus-Hankel matrices, *Numer. Math.* 52:665–682 (1988).

13   G. Heinig and K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Akademie-Verlag, Berlin, and Birkhäuser, 1984.

14   G. Heinig and K. Rost, On the inverses of Toeplitz-plus-Hankel matrices, *Linear Algebra Appl.* 106:39–52 (1988).

15   G. Heinig and K. Rost, Matrix representations of Toeplitz-plus-Hankel matrix inverses, *Linear Algebra Appl.* 113:65–78 (1989).

16   H. Lev-Ari, Factorization of Toeplitz plus Hankel matrices and other composite structured matrices, unpublished notes, May 1985.

17   I. Gohberg, T. Kailath, and I. Koltracht, Efficient solution of linear systems of equations with recursive structure, *Linear Algebra Appl.* 80:81–113 (1986).

18   A. E. Yagle, New analogs of split algorithms for arbitrary Toeplitz-plus-Hankel matrices, *IEEE Trans. Signal Process.* 39(1):2457–2463 (Nov. 1991).

19   P. Delsarte and Y. Genin, The split Levinson algorithm, *IEEE Trans. Acoust. Speech Signal Process.* 34(3):470–478 (June 1986).

20   Y. Bistritz, H. Lev-Ari, and T. Kailath, Immitance-type three-term Schur and Levinson recursions for quasi-Toeplitz complex Hermitian matrices, *SIAM J. Matrix Anal. Appl.* 12(3):497–520 (July 1991).

21   C. J. Zarowski, A Schur algorithm and linearly connected processor array for

Toeplitz-plus-Hankel matrices, *IEEE Trans. Signal Process.* 40(8):2065–2078 (Aug. 1992).

22 H. Lev-Ari and T. Kailath, Lattice filter parametrization and modeling of nonstationary processes, *IEEE Trans. Inform. Theory* 30(1):2–16 (Jan. 1984).

23 H. Lev-Ari and T. Kailath, Triangular factorization of structured Hermitian matrices, in *Oper. Theory Adv. Appl.* 18 (I. Gohberg, Ed.), Birkhäuser, Boston, 1986, pp. 301–324.

24 H. Lev-Ari, Extension problems under the displacement structure regime, in *Signal Processing, Part I: Signal Processing Theory* (L. Auslander, T. Kailath, and S. Mitter, Eds.), Springer-Verlag, New York, 1990, pp. 211–227.

25 A. H. Sayed, *Displacement Structure in Signal Processing and Mathematics*, Ph.D. Thesis, Stanford Univ., Stanford, Calif., Aug. 1992.

26 T. Kailath and A. H. Sayed, Fast algorithms for generalized displacement structures, in *Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems* (H. Kimura and S. Kodama, Eds.), Kobe, Japan, June 1991, Vol. 2, pp. 27–32.

27 T. Kailath, A theorem of I. Schur and its impact on modern signal processing, in *Oper. Theory Adv. Appl.* 18 (I. Gohberg, Ed.), Birkhäuser, Boston, 1986, pp. 9–30.

28 T. Kailath, Signal processing applications of some moment problems, in *Moments in Mathematics* (H. Landau, Ed.), Amer. Math. Soc., Providence, 1987, pp. 71–109.

29 I. Schur, Über potenzreihen die im innern des einheitskreises beschränkt sind, *J. Reine Angew. Math.* 147:205–232 (1917); English transl. in *Oper. Theory Adv. Appl.* 18, (I. Gohberg, Ed.), Birkhäuser, Boston, 1986, pp. 31–88.

30 Y. Bistritz and T. Kailath, Inversion and factorization of non-Hermitian quasi-Toeplitz matrices, *Linear Algebra Appl.* 98:77–121 (1988).

31 T. Kailath and J. Chun, Generalized displacement structure for block-Toeplitz, Toeplitz-block, and Toeplitz-derived matrices, *SIAM J. Matrix Anal. Appl.*, to appear.