The Lanczos Algorithm and Hankel Matrix Factorization

Daniel L. Boley

Department of Computer Science
University of Minnesota
Minneapolis, Minnesota

Tong J. Lee and Franklin T. Luk

School of Electrical Engineering
Cornell University
Ithaca, New York

Abstract

We explore the connections between the Lanczos algorithm for matrix tridiagonalization and two fast algorithms for Hankel matrix factorization. We show how the asymmetric Lanczos process is related to the Berlekamp-Massey algorithm, and how the symmetrized Lanczos process is related to the Phillips algorithm. We also discuss conditions under which the analysis of Phillips applies.

1 Introduction

In 1950 Lanczos [22] proposed a method for computing the eigenvalues of symmetric and nonsymmetric matrices. The idea was to reduce the given matrix to a tridiagonal form, from which the eigenvalues could be determined. A characterization of the breakdowns in the Lanczos algorithm in terms of algebraic conditions of controllability and observability was addressed in [6] and [26]. Hankel matrices arise in various settings, ranging from system identification [23] to algorithmic fault tolerance [4]. In his 1977 dissertation, Kung [20] studied the Berlekamp-Massey (BM, 1967) algorithm [1], [24] for solving Hankel equations, and remarked that their algorithm is related to the Lanczos process. There still exists strong interest in a simple exposition of the BM algorithm; see, e.g., [19] in 1989. In 1971 Phillips [28] proposed a Hankel triangularization scheme, and presented a derivation of his method using a special symmetrized Lanczos process with a weighted and possibly indefinite inner product. In this paper, we present the first systematic treatment of the connections between the Lanczos process and the two Hankel algorithms. We show how the BM and Phillips algorithms are just special cases of the asymmetric Lanczos and symmetrized Lanczos algorithms, respectively, using particular choices for the matrix and starting vectors. In addition, we point out an additional assumption, not mentioned in [28], that is essential for the application of the symmetrized Lanczos algorithm.

This paper is organized as follows. Section 2 presents the asymmetric and symmetrized versions of the Lanczos algorithm. In Section 3, the problem of orthogonalizing a sequence of polynomials is discussed, and it is shown how the Hankel matrix elements arise as *moments*. In Section 4, appropriate choices of matrices and vectors are made so that the two Lanczos schemes will compute two different factorizations of a Hankel matrix, just like the BM and Phillips algorithms. The paper concludes with a short numerical illustration and some remarks on the *breakdown* problem of the asymmetric Lanczos scheme.

2 Description of the Lanczos Process

We give a brief description of the Lanczos process. Consider a real vector space $\mathbf{V}$ with an associated weighted inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ of vectors $\mathbf{x}$ and $\mathbf{y}$ defined by

$$\langle \mathbf{x}, \mathbf{y} \rangle \;\equiv\; \mathbf{x}^{\mathrm{T}} W \mathbf{y}, \tag{2.1}$$

where W is some given real symmetric matrix. For $\mathbf{x} \neq \mathbf{y}$, we say that the vectors $\mathbf{x}$ and $\mathbf{y}$ are W*-orthogonal* if their inner product equals zero. Suppose that there exists an orthonormal basis $\mathbf{e}_1, \mathbf{e}_2, \cdots$, so that all the vectors in $\mathbf{V}$ can be expressed in terms of this basis:

$$\mathbf{x} = x_1 \mathbf{e}_1 + x_2 \mathbf{e}_2 + \cdots = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix}.$$

A linear operator will be denoted by a matrix A. We assume that the usual transpose A^{T} satisfies

$$\langle A^{\mathrm{T}} \mathbf{x}, \mathbf{y} \rangle \;=\; \langle \mathbf{x}, A \mathbf{y} \rangle,$$

thus requiring the condition that the matrices A and W *commute*. Note that in the usual asymmetric Lanczos algorithm, we have $W = I$ and so this commuting condition is satisfied automatically. In this paper we use a more general W in order to create a setting that encompasses both the usual asymmetric Lanczos algorithm and a modified symmetrized algorithm proposed by Phillips [28].

Although we may apply the Lanczos algorithm to possibly infinite vectors, we discuss the *nonsingularity* and *rank* of a matrix only for finite dimensional ones, so that we retain the usual definitions of these concepts. We represent vectors by the bold lowercase typeface $\mathbf{b}$, matrices by italic uppercase B, and linear spaces by bold uppercase $\mathbf{B}$. If

$$\mathbf{v}_k = A \mathbf{v}_{k-1} \quad \text{for all} \quad k,$$

the sequence of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \cdots\}$ is called a *Krylov sequence*, and the space spanned by these vectors is called the *right Krylov space* $\mathbf{G}$:

$$\mathbf{G} \equiv \mathit{span}(\mathbf{b}_1, A\mathbf{b}_1, A^2\mathbf{b}_1, \cdots).$$

We let $\mathbf{G}_k$ denote the truncated space generated by the matrix G_k:

$$G_k \equiv (\mathbf{b}_1, A\mathbf{b}_1, \cdots, A^{k-1}\mathbf{b}_1).$$

Likewise, we let $\mathbf{F}$ denote the *left Krylov space* :

$$\mathbf{F} \equiv \mathit{span}(\mathbf{c}_1, A^{\mathrm{T}}\mathbf{c}_1, (A^{\mathrm{T}})^2\mathbf{c}_1, \cdots),$$

and $\mathbf{F}_k$ the truncated space generated by the matrix F_k:

$$F_k \equiv (\mathbf{c}_1, A^{\mathrm{T}}\mathbf{c}_1, \cdots, (A^{\mathrm{T}})^{k-1}\mathbf{c}_1).$$

Note that F and G are used to distinguish between the leFt and riGht Krylov spaces.

2.1 Asymmetric Lanczos Method

Given a real matrix A and two real and non-null vectors $\mathbf{b}_1$ and $\mathbf{c}_1$ in $\mathbf{V}$, the asymmetric Lanczos algorithm generates two sequences of vectors:

$$B \equiv (\mathbf{b}_1, \mathbf{b}_2, \cdots) \quad \text{and} \quad C \equiv (\mathbf{c}_1, \mathbf{c}_2, \cdots),$$

such that

$$\mathit{span}(\mathbf{b}_1, \cdots, \mathbf{b}_k) = \mathbf{G}_k \quad \text{and} \quad \mathit{span}(\mathbf{c}_1, \cdots, \mathbf{c}_k) = \mathbf{F}_k, \quad \text{for all } k. \tag{2.2}$$

Given the $2k$ vectors $\mathbf{b}_1, \cdots, \mathbf{b}_k$ and $\mathbf{c}_1, \cdots, \mathbf{c}_k$, the two vectors $\mathbf{b}_{k+1}$ and $\mathbf{c}_{k+1}$ are computed by the formulae:

$$\mathbf{b}_{k+1} = A\mathbf{b}_k - (\mathbf{b}_1, \cdots, \mathbf{b}_k)\boldsymbol{\delta}_k$$

and

$$\mathbf{c}_{k+1} = A^{\mathrm{T}}\mathbf{c}_k - (\mathbf{c}_1, \cdots, \mathbf{c}_k)\boldsymbol{\gamma}_k,$$

for some coefficient vectors $\boldsymbol{\delta}_k$ and $\boldsymbol{\gamma}_k$. These vectors are chosen to enforce the W*-biorthogonality* conditions:

$$\langle \mathbf{b}_{k+1}, \mathbf{c}_i \rangle \;=\; 0 \quad \text{and} \quad \langle \mathbf{c}_{k+1}, \mathbf{b}_i \rangle \;=\; 0 \quad \text{for } i = 1, 2, \cdots, k, \tag{2.3}$$

for every k for which the $k \times k$ matrix

$$(\mathbf{c}_1, \cdots, \mathbf{c}_k)^{\mathrm{T}} W (\mathbf{b}_1, \cdots, \mathbf{b}_k), \quad \text{or equivalently } F_k^{\mathrm{T}} W G_k, \tag{2.4}$$

is nonsingular. In this section we make the *assumption* that $F_k^{\mathrm{T}} W G_k$ is nonsingular for all k (as long as $\mathbf{b}_k \neq 0$ and $\mathbf{c}_k \neq 0$), in which case only the last two entries of both $\boldsymbol{\delta}_k$ and $\boldsymbol{\gamma}_k$ are nonzero. When this assumption is relaxed, we face a *breakdown* problem that will be discussed in Section 5. This algorithm is the same as the nonsymmetric Lanczos algorithm found in [5], except that we are using a weighted inner product to enforce W-orthogonality, as opposed to an unweighted inner product to enforce ordinary orthogonality. We refer the reader to [5] for details of the algorithm, which we summarize as follows.

Algorithm AsymLanczos.

1. For $k = 1, 2, \cdots$ until stopped
2. Expand Krylov spaces: Set $\mathbf{b}_{k+1}^{(0)} = A\mathbf{b}_k$ and $\mathbf{c}_{k+1}^{(0)} = A^{\mathrm{T}}\mathbf{c}_k$.
3. Enforce the W-biorthogonality condition (2.3) by setting

$$\mathbf{b}_{k+1} = \mathbf{b}_{k+1}^{(0)} - (\mathbf{b}_1, \cdots, \mathbf{b}_k)\boldsymbol{\delta}_k,$$

and

$$\mathbf{c}_{k+1} = \mathbf{c}_{k+1}^{(0)} - (\mathbf{c}_1, \cdots, \mathbf{c}_k)\boldsymbol{\gamma}_k$$

via solving for the appropriate coefficients $\boldsymbol{\delta}_k$ and $\boldsymbol{\gamma}_k$. □

The process continues until either $\mathbf{b}_{r+1} = 0$ for some r, or $\mathbf{c}_{s+1} = 0$ for some s, whichever occurs first, although one could continue by appending zero vectors until one reaches zero vectors in both sequences. Hence the matrices B and C of generated vectors satisfy

$$AB = B\Delta \quad \text{and} \quad A^{\mathrm{T}}C = C\Gamma,$$

where Δ and Γ are tridiagonal matrices with unit subdiagonals, made up of the coefficient vectors $\boldsymbol{\delta}_k$ and $\boldsymbol{\gamma}_k$, for $k = 1, 2, \cdots$. The W-biorthogonality conditions (2.3) become

$$C^{\mathrm{T}} W B = D,$$

where D is a diagonal matrix. Note that in the case where $W = I$, the algorithm reduces to that given in Wilkinson [32, p. 388ff].

2.2 Symmetrized Lanczos Method

For the case of a real symmetric A, with the choice $\mathbf{b}_1 = \mathbf{c}_1$, one can show that

$$\mathbf{b}_i = \mathbf{c}_i \quad \text{for all } i.$$

The *W-biorthogonality* conditions (2.3) reduce to *W-orthogonality* conditions:

$$\langle \mathbf{b}_{k+1}, \mathbf{b}_i \rangle \;=\; 0 \quad \text{for } i = 1, 2, \cdots, k, \tag{2.5}$$

for every k for which the $k \times k$ matrix

$$(\mathbf{b}_1, \cdots, \mathbf{b}_k)^{\mathrm{T}} W (\mathbf{b}_1, \cdots, \mathbf{b}_k) \quad \text{or equivalently } G_k^{\mathrm{T}} W G_k, \tag{2.6}$$

is nonsingular. Under the assumption that (2.6) is nonsingular for all k (as long as $\mathbf{b}_k \neq 0$), we obtain a simplified symmetrized Lanczos process.

Algorithm SymmLanczos.

1. For $k = 1, 2, \cdots$ until stopped
2. Expand Krylov spaces: Set $\mathbf{b}_{k+1}^{(0)} = A\mathbf{b}_k$.
3. Enforce the W-orthogonality condition (2.5) by setting

$$\mathbf{b}_{k+1} = \mathbf{b}_{k+1}^{(0)} - (\mathbf{b}_1, \cdots, \mathbf{b}_j)\boldsymbol{\delta}_k$$

via solving for the appropriate coefficients $\boldsymbol{\delta}_k$. □

The process continues until $\mathbf{b}_{r+1} = 0$ for some r. Thus the matrix B of generated vectors satisfies the two conditions:

$$AB = B\Delta \quad \text{and} \quad B^{\mathrm{T}} W B = D,$$

where Δ is tridiagonal and D is diagonal. Note that, under our assumption that $G_k^{\mathrm{T}} W G_k$ is nonsingular for all k, Algorithm SymmLanczos is same as the method given in Phillips [28]. If in addition $W = I$, the algorithm reduces to the usual symmetric Lanczos algorithm [13, p. 485ff], [32, p. 394ff], in which case the nonsingularity of (2.6) is guaranteed for all k as long as $\mathbf{b}_k \neq 0$.

3 Application to Sequences of Polynomials

Assume there exists a real-valued *inner product* $\langle f, g\rangle$ which satisfies the usual properties except for positivity:

$$\langle f, \alpha g + h\rangle = \langle \alpha g + h, f\rangle = \alpha\langle g, f\rangle + \langle h, f\rangle$$

and

$$\langle xf, g\rangle = \langle f, xg\rangle,$$

for any real-valued functions f, g, h of x. Given a sequence of polynomials $p_0, p_1, \cdots$, of exact degrees, and the *moments*:

$$\eta_k = \langle p_k, p_0\rangle, \quad \text{for } k = 0, 1, 2, \cdots, \tag{3.1}$$

where p_0 is a constant polynomial, we wish to find another sequence of polynomials $q_0, q_1, \cdots$, of exact degrees so that the q's are *orthogonal* with respect to the given inner product. Note that the relation defining the inner product may *not* be known. For the case of an *ordinary* inner product, viz., $\langle f, f\rangle > 0$ for all nonzero f, the problem has been extensively studied in the literature; see, e.g., [11] and references therein. However, only recently has this problem been addressed for more general inner products; see, e.g., the *modified Chebyshev algorithm* in [12]. This problem was addressed in [5] for the case where the inner product is a discrete sum over a finite number of knots.

3.1 Asymmetric Orthogonalization

Since the polynomials are of exact degrees they obey the recurrence formulae:

$$x\mathbf{p}^{\mathrm{T}} = \mathbf{p}^{\mathrm{T}} Z_p \quad \text{and} \quad x\mathbf{q}^{\mathrm{T}} = \mathbf{q}^{\mathrm{T}} Z_q,$$

where

$$\mathbf{p} = \begin{pmatrix} p_0(x) \\ p_1(x) \\ p_2(x) \\ \vdots \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} q_0(x) \\ q_1(x) \\ q_2(x) \\ \vdots \end{pmatrix},$$

and both Z_p and Z_q are unreduced infinite upper Hessenberg matrices. The polynomials are also related by an infinite upper triangular matrix of coefficients B:

$$\mathbf{q}^{\mathrm{T}} = \mathbf{p}^{\mathrm{T}} B.$$

From the definitions, we have that

$$\mathbf{p}^{\mathrm{T}} B Z_q = x\mathbf{p}^{\mathrm{T}} B = \mathbf{p}^{\mathrm{T}} Z_p B, \tag{3.2}$$

and thus

$$B Z_q = Z_p B. \tag{3.3}$$

We are interested in further exploring the relations between the two sequences. We make the simplifying assumption that the zero degree polynomials are scaled so that $p_0 = q_0$. The upper Hessenberg structure of Z_p implies, among other things, that for every k,

$$\mathit{span}(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_k) = \mathit{span}(\mathbf{b}_1, Z_p\mathbf{b}_1, \cdots, Z_p^{k-1}\mathbf{b}_1), \tag{3.4}$$

where $\mathbf{b}_i$ denotes the i-th column of B.

Define the matrix C of *mixed moments*:

$$c_{ij} = \langle p_{i-1}, q_{j-1} \rangle, \quad \text{for } i,j = 1,2,3,\cdots. \tag{3.5}$$

Let $\mathbf{c}_i$ denote the i-th column of C. Since $q_0 = p_0$, the first column $\mathbf{c}_1$ of C is given by

$$\mathbf{c}_1 = \begin{pmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \\ \vdots \end{pmatrix}.$$

Following [12], we use the extended notation that

$$C \equiv \langle \mathbf{p}, \mathbf{q}^{\mathrm{T}} \rangle,$$

where the inner product applied to a vector of functions means that it is applied individually to each element. By linearity we have that

$$Z_p^{\mathrm{T}} \langle \mathbf{p}, \mathbf{q}^{\mathrm{T}} \rangle = \langle x\mathbf{p}, \mathbf{q}^{\mathrm{T}} \rangle = \langle \mathbf{p}, x\mathbf{q}^{\mathrm{T}} \rangle = \langle \mathbf{p}, \mathbf{q}^{\mathrm{T}} \rangle Z_q \tag{3.6}$$

Equation (3.6) reduces to

$$Z_p^{\mathrm{T}} C = C Z_q \tag{3.7}$$

As with the B matrix, this implies that for all k

$$span(\mathbf{c}_1, \cdots, \mathbf{c}_k) = span(\mathbf{c}_1, Z_p^{\mathrm{T}} \mathbf{c}_1, \cdots, (Z_p^{\mathrm{T}})^{k-1} \mathbf{c}_1). \tag{3.8}$$

Let us discuss some specific choices for the polynomials. There are two common choices for p's. One, they are chosen as the monomials, which is equivalent to Z_p being a *shift* matrix. Two, the p's are generated via a three-term recurrence, equivalent to Z_p being a tridiagonal matrix of recurrence coefficients. Independently of the choice for the p's, we can make arbitrary choices for the q's. If in particular we choose the q's to be *orthogonal* with respect to $\langle \cdot, \cdot \rangle$, then the corresponding matrix condition is that the matrix

$$D = \langle \mathbf{q}, \mathbf{q}^{\mathrm{T}} \rangle = \langle \mathbf{q}, \mathbf{p}^{\mathrm{T}} \rangle B = C^{\mathrm{T}} B \tag{3.9}$$

be diagonal. We then observe that the conditions (3.3), (3.7), (3.9) and the Krylov sequence conditions (3.4) and (3.8) exactly match the properties of the vectors generated by the Lanczos process when started with the matrix Z_p and right vector $\mathbf{b}_1 = \mathbf{e}_1$ and left vector $\mathbf{c}_1$ composed of the moments. It follows that if such a sequence of orthogonal q's exist, then the vectors generated by the Lanczos process will satisfy (3.9), and vice versa.

We now discuss the computation of the leading finite-dimensional part of the above infinite vectors. Suppose we are given only the first $2n-1$ moments $\eta_0, \eta_1, \cdots, \eta_{2n-2}$ as well as the leading $(2n-1) \times (2n-1)$ part of Z_p, which we henceforth call Z for simplicity. Because of the lower Hessenberg form of Z^{T}, we know the first $2n-2$ entries in $Z^{\mathrm{T}} \mathbf{c}_1$, the first $2n-3$ components of $(Z^{\mathrm{T}})^2 \mathbf{c}_1$, and so on. Thus, we will know the *leading anti-triangle* of the left Krylov matrix:

$$F_n = span(\mathbf{c}_1, Z_p^{\mathrm{T}} \mathbf{c}_1, (Z_p^{\mathrm{T}})^2 \mathbf{c}_1, \cdots, (Z_p^{\mathrm{T}})^{n-1} \mathbf{c}_1), \tag{3.10}$$

Note that the leading $n \times n$ principal submatrix of F_n is known. Recall also the right Krylov matrix:

$$G_n = span(\mathbf{b}_1, Z_p \mathbf{b}_1, Z_p^2 \mathbf{b}_1, \cdots, Z_p^{n-1} \mathbf{b}_1).$$

The two sequences of vectors $\{\mathbf{b}_i\}$ and $\{\mathbf{c}_i\}$ satisfying (3.9) can be generated by applying an oblique Gram-Schmidt process to F_n and G_n. Due to the upper triangular nature of the vectors $\mathbf{b}_i$, the conditions (3.9) for the first n vectors involve only the first n entries of both the $\mathbf{b}$ and $\mathbf{c}$ vectors.

The Lanczos process will generate a sequence of vectors $\mathbf{b}_1, \mathbf{b}_2, \cdots$, and $\mathbf{c}_1, \mathbf{c}_2, \cdots$. With the first $2n-1$ entries of $\mathbf{c}_1$ known and $\mathbf{b}_1 = \mathbf{e}_1$, the Lanczos algorithm will generate at least the vectors $\mathbf{b}_1, \cdots, \mathbf{b}_n$ and leading n entries of $\mathbf{c}_1, \cdots, \mathbf{c}_n$. Each polynomial q_k will be defined in terms of the originally given set of p polynomials by the relation:

$$q_k(x) = \mathbf{p}^{\mathrm{T}}(x)\mathbf{b}_{k+1}, \quad \text{for } k = 0, 1, 2, \cdots.$$

The moments involving q_k are the entries of $\mathbf{c}_{k+1}$:

$$\mathbf{c}_{k+1} = \begin{pmatrix} \langle p_0, q_k \rangle \\ \langle p_1, q_k \rangle \\ \langle p_2, q_k \rangle \\ \vdots \end{pmatrix} \tag{3.11}$$

If k is an index such that the matrix $(\mathbf{c}_1, \cdots, \mathbf{c}_k)^{\mathrm{T}}(\mathbf{b}_1, \cdots, \mathbf{b}_k)$ is nonsingular, then $\mathbf{c}_{k+1}$ will orthogonal to $\mathbf{b}_i$, for $i = 1, 2, \cdots, k$. Due to the upper triangularity of B, this means that the first k entries of (3.11) will be zero, and so C will be lower triangular from the diagonality condition (3.9). Note that this is a condition involving only finitely many leading entries of (3.11). So for such indices k, the polynomial q_k will be orthogonal to all polynomials p's of degrees lower than k.

We also note that in general, it is well known that there is a loss of bi-orthogonality among the Lanczos vectors generated. But in our situation, the bi-orthogonality conditions on the vectors $\mathbf{c}_1, \mathbf{c}_2, \cdots$ translate directly into the condition that the generated matrix C be lower triangular, a condition that maintained numerically almost automatically.

3.2 Symmetrized Orthogonalization

Assume that the moments matrix H has a triangular decomposition:

$$H = R^{\mathrm{T}} D R,$$

where R is unit upper triangular and D is diagonal. To run the Lanczos process, we choose some initial matrix A that is symmetric, and choose the same initial vectors: $\mathbf{b}_1 = \mathbf{c}_1$. Hence the left and right Krylov matrices will be identical. Assume that the Krylov matrix G_n is nonsingular, and define a matrix B by

$$B \equiv G_n R^{-1}. \tag{3.12}$$

It can be shown that the columns of B are generated by the symmetrized Lanczos process, and that they satisfy the relations:

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \quad \text{for } i \neq j,$$

and

$$\langle \mathbf{b}_i, \mathbf{b}_i \rangle = D_{i,i},$$

where $\mathbf{b}_i$ denotes the i-th column of B, and $D_{i,i}$ denotes the (i,i) element of D. Most importantly, we can now find the matrix R from (3.12):

$$R^{-1} = G_n^{-1} B. \tag{3.13}$$

Define new polynomials q's from the p's via the formula:

$$\mathbf{q}^{\mathrm{T}} = \mathbf{p}^{\mathrm{T}} R^{-1}.$$

So we get

$$\langle \mathbf{q}, \mathbf{q}^{\mathrm{T}} \rangle = R^{-\mathrm{T}} \langle \mathbf{p}, \mathbf{p}^{\mathrm{T}} \rangle R^{-1} = R^{-\mathrm{T}} H R^{-1} = D,$$

verifying that the polynomials q's are orthogonal with respect to the given inner product, just as we desire.

4 Factorization of a Hankel Matrix

Hankel solvers and the closely related Toeplitz solvers have been studied for a long time; see, e.g., [13]. Most Toeplitz solvers are based on the *shift invariance* of a Toeplitz matrix, for which a principal leading submatrix is identical to the principal trailing submatrix of the same size. The resulting methods can compute the LU factors of a Toeplitz matrix in $O(n^2)$ operations, but require that all leading principal submatrices be nonsingular. We will refer to a matrix with all nonsingular leading principal submatrices as *strongly nonsingular*. The Lanczos recursion leads to fast Hankel factorizers which are equivalent in cost to the fast Toeplitz factorizers. The resulting recursion formulae to factorize a strongly nonsingular Hankel matrix have appeared in several papers under different guises, going all the way back to Chebyshev [8]. Early algorithms for factorizing a Hankel matrix appeared in [29], [30], [31], and the connection with the Lanczos algorithm either on a nonsymmetric matrix or using an *indefinite* inner product appeared in [14], [18], [20], [28]. When the permuted Yule-Walker equations (a Hankel system with a special right hand side) are solved, the resulting method essentially computes the LU factors of the Hankel matrix [1], [24], [25]. The same sets of equations arise in identification problems, where we would like to construct the transfer function from the impulse response of a dynamical system [23]. The relation between the asymmetric Lanczos process and fast Hankel factorization and/or inversion algorithms has been explored more recently in [3], [4], [16], [17], [21].

In the next two subsections, we apply the Lanczos process to define two procedures for decomposing an $n \times n$ strongly nonsingular Hankel matrix

$$H = \begin{pmatrix} \eta_0 & \eta_1 & \eta_2 & \cdots & \eta_{n-1} \\ \eta_1 & \eta_2 & \eta_3 & \cdots & \eta_n \\ \eta_2 & \eta_3 & \eta_4 & \cdots & \eta_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \eta_{n-1} & \eta_n & \eta_{n+1} & \cdots & \eta_{2n-2} \end{pmatrix}.$$

To be specific, the asymmetric Lanczos process computes the factorization:

$$HU = L, \tag{4.1}$$

where the matrix U is unit upper triangular, and the matrix L is lower triangular, and the symmetrized Lanczos process calculates the factorization:

$$H = R^{\mathrm{T}} D R, \tag{4.2}$$

where the matrix R is unit upper triangular, and the matrix D is diagonal. From the uniqueness of the triangular decomposition of H we conclude that $L^{\mathrm{T}} = DR$.

4.1 Asymmetric Lanczos Factorization

We define two $(2n-1)$-element vectors $\mathbf{b}_1$ and $\mathbf{c}_1$. The former is the first coordinate unit vector, and the latter contains the parameters generating the Hankel matrix H. That is,

$$\mathbf{b}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \equiv \mathbf{e}_1 \quad \text{and} \quad \mathbf{c}_1 = \begin{pmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_{2n-2} \end{pmatrix}.$$

The weighting matrix W is chosen as the identity matrix of order $2n-1$:

$$W = I_{2n-1}.$$

Define a $(2n-1) \times (2n-1)$ *shift-down* matrix:

$$Z = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix},$$

and let

$$A = Z.$$

Then the left Krylov sequence generated by $\mathbf{c}_1$ has the form

$$F_n = (\, \mathbf{c}_1,\ Z^{\mathrm{T}}\mathbf{c}_1, \cdots,\ (Z^{\mathrm{T}})^{n-1}\mathbf{c}_1 \,) \equiv \begin{pmatrix} H \\ K \end{pmatrix}, \tag{4.3}$$

where K is an $(n-1) \times n$ upper anti-triangular matrix of the excess parameters, with Hankel structure.

The right Krylov space, for any k,

$$\mathit{span}(\mathbf{b}_1, Z\mathbf{b}_1, \cdots, Z^{k-1}\mathbf{b}_1) = \mathit{span}(\mathbf{e}_1, \cdots, \mathbf{e}_k)$$

spans the same space as the first k columns of the identity matrix. The condition that

$$(\mathbf{c}_1, Z^{\mathrm{T}}\mathbf{c}_1, \cdots, (Z^{\mathrm{T}})^{k-1}\mathbf{c}_1)^{\mathrm{T}}(\mathbf{b}_1, Z\mathbf{b}_1, \cdots, Z^{k-1}\mathbf{b}_1) \equiv F_k^{\mathrm{T}} G_k \tag{4.4}$$

be nonsingular is equivalent to the condition that the leading $k \times k$ principal submatrix of H is nonsingular. Let us first consider the case where this holds for every k. Then the vectors $C = (\mathbf{c}_1, \cdots, \mathbf{c}_n)$ will be lower triangular, from the bi-orthogonality condition. Since the leading k columns of C span the same space as F_k, for every k, it follows that

$$C = F_n U \tag{4.5}$$

for some upper triangular matrix U. If we denote by L the first n rows of C, then the first n rows of (4.5) gives the desired factorization of H as in (4.1).

We note that this process is equivalent to generating the sequence of polynomials orthogonal with respect to an ordinary inner product $\langle \cdot, \cdot \rangle$ whose moments derived from some polynomials p's are the given

Hankel parameters $\mathbf{c}_1$. The Krylov sequence (4.3) equals (3.10), and the lower triangular matrix C equals the mixed moments matrix of (3.5), which is lower triangular when the polynomials being generated are orthogonal with respect to the inner product.

Unfortunately, the matrix U is not generated by the Lanczos method. However, we may generate it by recording the operations that go into generating C. With this particular choice of starting information, the Lanczos process amounts to just reducing the left Krylov sequence F_n to lower triangular form C by means of *column operations*. In other words, each column of C is obtained by applying Z^{T} (i.e., shifting up) and then subtracting multiples of previous columns to reduce it to lower triangular form. That is,

$$\theta_j \mathbf{c}_{j+1} = Z^{\mathrm{T}} \mathbf{c}_j - (\mathbf{c}_1, \cdots, \mathbf{c}_j) \boldsymbol{\gamma}_j \tag{4.6}$$

for some vector γ_j and some scalar θ_j. From (4.6) we have that

$$\mathbf{c}_j = K \mathbf{u}_j,$$

where $\mathbf{u}_j$ is the j-th column of U. Then we have the following identity from the Hankel structure of K:

$$Z^{\mathrm{T}} \mathbf{c}_j = Z^{\mathrm{T}} K \mathbf{u}_j = K Z \mathbf{u}_j.$$

Thus, we may express $\mathbf{c}_{j+1} = K \mathbf{u}_{j+1}$, where

$$\theta_j \mathbf{u}_{j+1} = Z \mathbf{u}_j - (\mathbf{u}_1, \cdots, \mathbf{u}_j) \boldsymbol{\gamma}_j \tag{4.7}$$

and the coefficients $\boldsymbol{\gamma}_j$ and θ_j are those defined in (4.6). Thus, as we perform the up-shifting and column operations to generate the $\mathbf{c}$ vectors, we perform down-shifting and the same column operations on the $\mathbf{u}$ vectors to generate the U matrix. Under the usual situation where (4.4) is nonsingular, the vector $\boldsymbol{\gamma}_j$ have only two nonzero entries, and hence each $\mathbf{u}_j$ can be generated with only $O(j)$ operations.

We summarize the process with the following procedure.

Algorithm AsymHankel

1. For $j = 1, 2, \cdots, n-1$
2. $\mathbf{c}_{j+1}^{(0)} = Z^{\mathrm{T}} \mathbf{c}_j$.
3. $\mathbf{u}_{j+1}^{(0)} = Z \mathbf{u}_j$.
4. $\theta_j \mathbf{c}_{j+1} = \mathbf{c}_{j+1}^{(0)} - (\mathbf{c}_{j-1}, \mathbf{c}_j) \begin{pmatrix} \gamma_{j-1,j} \\ \gamma_{jj} \end{pmatrix}$, where the coefficients $\gamma_{j-1,j}$ and γ_{jj} are computed to annihilate the entries $(\mathbf{c}_{j+1}^{(0)})_{j-1}$ and $(\mathbf{c}_{j+1}^{(0)})_j$.
5. $\theta_j \mathbf{u}_{j+1} = \mathbf{u}_{j+1}^{(0)} - (\mathbf{u}_{j-1}, \mathbf{u}_j) \begin{pmatrix} \gamma_{j-1,j} \\ \gamma_{jj} \end{pmatrix}$. $\square$

Thus, at the j-th stage, we augment the matrix $(\mathbf{c}_1, \cdots, \mathbf{c}_j)$ to obtain

$$(\mathbf{c}_1, \cdots, \mathbf{c}_j, \mathbf{c}_{j+1}^{(0)}) = \begin{pmatrix} c_{11} & & & & & \\ c_{21} & c_{22} & & & & \\ \vdots & \vdots & \ddots & & & \\ c_{j-1,1} & c_{j-1,2} & \cdots & c_{j-1,j-1} & & c_{j-1,j+1}^{(0)} \\ c_{j1} & c_{j2} & \cdots & c_{j,j-1} & c_{jj} & c_{j,j+1}^{(0)} \\ c_{j+1,1} & c_{j+1,2} & \cdots & c_{j+1,j-1} & c_{j+1,j} & c_{j+1,j+1}^{(0)} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \end{pmatrix}$$

and then the two entries $c_{j-1,j+1}^{(0)}$ and $c_{j,j+1}^{(0)}$ are annihilated. In detail, steps 4 and 5 above can be written out as

4.1 $\gamma_{j-1,j} = \frac{c_{j-1,j+1}^{(0)}}{c_{j-1,j-1}} = \frac{c_{jj}}{c_{j-1,j-1}}.$

4.2 $\mathbf{c}_{j+1}^{(1)} = \mathbf{c}_{j+1}^{(0)} - \gamma_{j-1,j}\mathbf{c}_{j-1}.$

5.2 $\mathbf{u}_{j+1}^{(1)} = \mathbf{u}_{j+1}^{(0)} - \gamma_{j-1,j}\mathbf{u}_{j-1}.$

4.3 $\gamma_{jj} = \frac{c_{j,j+1}^{(1)}}{c_{jj}} = \frac{c_{j+1,j}}{c_{jj}} - \frac{c_{j,j-1}}{c_{j-1,j-1}}.$

4.4 $\mathbf{c}_{j+1} = \mathbf{c}_{j+1}^{(1)} - \gamma_{jj}\mathbf{c}_j.$

5.4 $\mathbf{u}_{j+1} = \mathbf{u}_{j+1}^{(1)} - \gamma_{jj}\mathbf{u}_j.$

The reader will recognize from the formulae in steps 4.1 and 4.3 that this is the same as the Chebyshev algorithm [8, *Oeuvres*, p. 482], [11] in the theory of moments and orthogonal polynomials, as well as the Berlekamp-Massey algorithm in coding [1], [19], [24].

4.2 Symmetrized Lanczos Factorization

In this section we consider the factorization of a strongly nonsingular Hankel matrix H. Such a matrix has the decomposition as given by (4.2), and in this section we derive the algorithm of Phillips [28] to generate this decomposition. Phillips [28] showed how H can be viewed as a moments matrix. Consider the right Krylov matrix G_n:

$$G_n = (\mathbf{b}_1, A\mathbf{b}_1, \cdots, A^{n-1}\mathbf{b}_1),$$

where A is some symmetric matrix and $\mathbf{b}_1$ is some vector such that G_n is nonsingular. Define the inner product (2.1) with a symmetric weighting matrix W defined by

$$W = G_n^{-\mathrm{T}} H G_n^{-1}. \tag{4.8}$$

Then the (i,j) element of H satisfies

$$(H)_{i,j} = \eta_{i+j-2} = \langle A^{i-1}\mathbf{b}_1, A^{j-1}\mathbf{b}_1 \rangle,$$

as long as A and W commute. The following development depends on finding matrices A and W which commute, but as will be seen in Section 4.3, a W defined by (4.8) may *not* always commute with A. So, in general, can we always find two symmetric matrices A and W that commute for any given H? The answer is yes. At the end of this section, we give one example of such A and W for any given strongly nonsingular H. Hence the following development is not vacuous.

We can modify the symmetric Lanczos tridiagonalization process (cf. [13, p. 476ff]) to generate a matrix B with W-orthogonal columns that satisfies

$$AB = BT, \tag{4.9}$$

where T is tridiagonal. Let

$$B = (\,\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n\,).$$

The columns of B are determined by using the formula:

$$\mathbf{b}_{i+1} = (A - \alpha_i I)\mathbf{b}_i - \beta_i \mathbf{b}_{i-1}, \quad \text{for } i = 1, \cdots, n-1, \tag{4.10}$$

where α_i and β_i are chosen so $\mathbf{b}_{i+1}$ is W-orthogonal to $\mathbf{b}_i$ and $\mathbf{b}_{i-1}$, and hence also to $\mathbf{b}_{i-2}, \cdots, \mathbf{b}_1$. The tridiagonal matrix T is defined by

$$T \equiv (1, \alpha_i, \beta_{i+1}).$$

By analogy with the symmetric Lanczos process, the matrix B forms a part of the W-orthogonal QR decomposition of the Krylov matrix G_n:

$$G_n = BR,$$

where B has W-orthogonal columns which are scaled so that R is unit upper triangular. Hence B satisfies

$$B^T W B = D. \tag{4.11}$$

Now, consider the Krylov matrix G_{2n}, given by

$$G_{2n} = (\mathbf{b}_1, A\mathbf{b}_1, \cdots, A^{2n-1}\mathbf{b}_1) \equiv (G_n, \hat{G}).$$

i.e.,

$$\hat{G} \equiv (A^n \mathbf{b}_1, A^{n+1}\mathbf{b}_1, \cdots, A^{2n-1}\mathbf{b}_1) = A^n G_n.$$

Let Z_L denote a $2n \times 2n$ *left-shift* matrix:

$$Z_L \equiv \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix}.$$

Note that Z_L has the same structure as the order $2n-1$ *down-shift* matrix Z of Section 4.1. We have

$$(AG_{2n})_j = (G_{2n} Z_L)_j, \quad \text{for } j = 1, \cdots, 2n-1,$$

where M_j denotes the j-th column of a matrix M. Since

$$AG_{2n} = AB(R, B^{-1}\hat{G}) = BT(R, B^{-1}\hat{G})$$

and

$$G_{2n} Z_L = B(R, B^{-1}\hat{G}) Z_L,$$

we get

$$\left(T(R, B^{-1}\hat{G})\right)_j = \left((R, B^{-1}\hat{G}) Z_L\right)_j, \quad \text{for } j = 1, \cdots, 2n-1,$$

i.e., a *shift-to-left* operation on G_{2n} is equivalent to a three-term recurrence on R. However, it is expensive to compute T, and a better way is to express T in terms of R. Define the tridiagonal matrix

$$P \equiv DTD^{-1}$$

and

$$C \equiv (DR, DB^{-1}\hat{G}). \tag{4.12}$$

The first n columns of C give us the two desired matrices D and R. Then

$$PC = DT(R, B^{-1}\hat{G})$$

and so

$$(PC)_j = (CZ_L)_j, \quad \text{for } j = 1, \cdots, 2n-1.$$

Since P is tridiagonal, the above formula shows that the rows of C obey a recurrence, which with some algebraic manipulation can be written as [28]:

$$c_{i+1,j} = c_{i,j+1} - \alpha_i c_{i,j} - \beta_i c_{i-1,j},$$

where

$$\alpha_i = \frac{c_{i,i+1}}{c_{i,i}} - \frac{c_{i-1,i}}{c_{i-1,i-1}} \quad \text{and} \quad \beta_i = \frac{c_{i,i}}{c_{i-1,i-1}}.$$

To find the initial conditions, consider

$$\mathbf{e}_1^{\mathrm{T}} C = \mathbf{e}_1^{\mathrm{T}} R^* C = \mathbf{e}_1^{\mathrm{T}} (R^{\mathrm{T}} DR, R^{\mathrm{T}} DB^{-1} \hat{G}) = \mathbf{e}_1^{\mathrm{T}} (H, G_n^{\mathrm{T}} W A^n G_n),$$

since

$$R^{\mathrm{T}} DB^{-1} = HR^{-1}B^{-1} = HG_n^{-1} = G_n^{\mathrm{T}} W.$$

So the first row of C is given by

$$c_{1,j} = \eta_{j-1}, \quad \text{for } j = 1, \cdots, 2n-1. \tag{4.13}$$

Phillips [28] has thus developed an ingenious row recurrence scheme that does not require any explicit knowledge of A and W. His procedure is summarized as follows.

Algorithm SymmHankel

1. For $i = 1, 2, \cdots, n-1$
2. $\quad \alpha_i = \frac{c_{i,i+1}}{c_{i,i}} - \frac{c_{i-1,i}}{c_{i-1,i-1}}; \;\; \alpha_1 = \frac{c_{1,2}}{c_{1,1}}$.
3. $\quad \beta_i = \frac{c_{i,i}}{c_{i-1,i-1}}; \;\; \beta_1 = 0$.
4. $\quad$ For $j = i+1, i+2, \cdots, 2n-1-i$
5. $\quad\quad c_{i+1,j} = c_{i,j+1} - \alpha_i c_{i,j} - \beta_i c_{i-1,j}$. □

We note that this process is equivalent to generating the sequence of polynomials q's orthogonal with respect to an indefinite inner product whose moments derived from the polynomials p's are the given Hankel parameters $\mathbf{c}_1$. Not surprisingly, this algorithm computes parameters α_i and β_i that are same as the $\gamma_{i,i}$ and $\gamma_{i-1,i}$ of Algorithm AsymHankel. The explanation is that while the latter algorithm computes the factor L column-wise, Algorithm SymmHankel computes the product DR row-wise, and that $L^{\mathrm{T}} = DR$.

Even though the specific A and W are not needed to carry out Algorithm SymmHankel, the derivation of the algorithm depends on the existence of symmetric A and W related by (4.8) which commute. To construct one such example, suppose that the entries of H are moments with respect to some set of weights $\{\zeta_i\}$ and knots $\{x_i\}$, for $i = 0, \cdots, n-1$, in the inner product defined by

$$\langle p, q \rangle = \sum_{0}^{n-1} p(x_i) q(x_i) \zeta_i.$$

For example, in checksum-based error correction schemes, the entries of H are the moments of an indefinite inner product of known knots and unknown weights that play the role of errors [4]. This corresponds to the

Reed-Solomon code [2] and the Berlekamp-Massey Algorithm. If the knots are unknown, they may often be computed as the roots of the n-th degree orthogonal polynomial generated by Algorithm AsymLanczos, as described in [5]. Once the knots are known, the weights $\{\zeta_i\}$ are the solution to a square nonsingular Vandermonde system [4,5].

Given this set of knots and weights, the entries of the Hankel matrix H can be written as

$$h_{ij} = \eta_{i+j-2} = \sum_{k=0}^{n-1} x_k^{i-1} \zeta_k x_k^{j-1},$$

yielding the matrix relation

$$H = VYV^{\mathrm{T}}, \tag{4.14}$$

where the matrix V is Vandermonde and matrix Y is diagonal:

$$V^{\mathrm{T}} = \begin{pmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{pmatrix} \quad \text{and} \quad Y = \begin{pmatrix} \zeta_0 & 0 & \cdots & 0 \\ 0 & \zeta_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \zeta_{n-1} \end{pmatrix}.$$

We *emphasize* here that neither the sequence $\{x_i\}$ nor the $\{\zeta_i\}$ are required for the actual computation, they are useful only for our theoretical discussion.

Finally we choose

$$A = \begin{pmatrix} x_0 & 0 & \cdots & 0 \\ 0 & x_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_{n-1} \end{pmatrix} \quad \text{and} \quad \mathbf{b}_1 = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

so that

$$G_n = V^{\mathrm{T}}$$

and

$$W = V^{-1} H V^{-\mathrm{T}} = Y.$$

Since both matrices A and W are diagonal, they commute.

4.3 Work of Phillips

Phillips' elegant analysis [28] works only if A and W commute. For example, to ensure that $\mathbf{b}_{i+1}$ is W-orthogonal to $\mathbf{b}_{i-1}$, he used

$$\beta_i = \frac{\langle \mathbf{b}_i, \mathbf{b}_i \rangle}{\langle \mathbf{b}_{i-1}, \mathbf{b}_{i-1} \rangle}, \quad \text{for } i = 2, \cdots, n, \tag{4.15}$$

which should be replaced by

$$\beta_i = \frac{\langle \mathbf{b}_{i-1}, A\mathbf{b}_i \rangle}{\langle \mathbf{b}_{i-1}, \mathbf{b}_{i-1} \rangle}, \quad \text{for } i = 2, \cdots, n. \tag{4.16}$$

Nonetheless, (4.15) and (4.16) are equivalent under the assumption of commutativity, in which case (4.15) is actually a better numerical formula. Now, let

$$H = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 3 & 2 & 1 \end{pmatrix}, \qquad \mathbf{b}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{pmatrix}. \tag{4.17}$$

Following [28] we calculate

$$G_n = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \end{pmatrix} \quad \text{and} \quad W = \begin{pmatrix} 0.5556 & -1.5000 & 0.6111 \\ -1.5000 & 4.0000 & -1.0000 \\ 0.6111 & -1.0000 & 0.2222 \end{pmatrix}$$

Note that

$$AW - WA = \begin{pmatrix} 0 & 1.5000 & -1.8333 \\ -1.5000 & 0 & 2.0000 \\ 1.8333 & -2.0000 & 0 \end{pmatrix} \neq 0.$$

Via (4.15) we get

$$B = \begin{pmatrix} 1 & -1 & 2 \\ 1 & 0 & 1 \\ 1 & 2 & 5 \end{pmatrix} \quad \text{and} \quad T = \begin{pmatrix} 2 & -1 & 0 \\ 1 & 2 & -8 \\ 0 & 1 & 3 \end{pmatrix}.$$

Finally,

$$AB - BT = \begin{pmatrix} 0 & 0 & -12.0000 \\ 0 & 0 & -1.0000 \\ 0 & 0 & 21.0000 \end{pmatrix},$$

and the key equation (4.9) fails to hold. Using (4.16) instead of (4.15) would also give us bad answers in that

$$AB - BT = \begin{pmatrix} 0 & 0 & -1.0000 \\ 0 & 0 & -1.0000 \\ 0 & 0 & -1.0000 \end{pmatrix}.$$

5 Numerical Illustration

We use the numerical example in the previous section to illustrate the algorithms. Let H be defined by (4.17). Then the left Krylov sequence (4.3) is

$$F_3 = (\, \mathbf{c}_1,\ Z^{\mathrm{T}}\mathbf{c}_1\ (Z^{\mathrm{T}})^2\mathbf{c}_1 \,) \equiv \begin{pmatrix} H \\ K \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 3 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

The algorithm AsymHankel will generate the factorization $C = F_3U$ (4.5) where

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 2 & -1 & 0 \\ 3 & -4 & 8 \\ 2 & -3 & 6 \\ 1 & -2 & 5 \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} 1 & -2 & 5 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{pmatrix},$$

yielding the final factorization of the original matrix H as

$$C \equiv \begin{pmatrix} 1 & 0 & 0 \\ 2 & -1 & 0 \\ 3 & -4 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 5 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{pmatrix} \equiv HU$$

The algorithm SymHankel will generate the rows

$$\{c_{i,j}\} = \begin{pmatrix} 1 & 2 & 3 & 2 & 1 \\ 0 & -1 & -4 & -3 & \times \\ 0 & 0 & 8 & \times & \times \end{pmatrix},$$

where the subdiagonal zero entries and the entries marked "×" are not computed. From (4.12) we obtain DR and then D and R by scaling the rows of DR to have unit diagonal entries:

$$DR = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -4 \\ 0 & 0 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 8 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}.$$

The decomposition of H is then $H = R^{\mathrm{T}} DR$. Note that we do not need to know explicitly the commuting matrices A and W.

6 Final Remarks

In this paper, we have shown how two well known fast Hankel factorization methods can be viewed as special cases of the Lanczos algorithm. For simplicity in presentation, we have avoided the *breakdown* problem for the asymmetric Lanczos algorithm. The problem has been considered in [5], [7], [15], [27], with an approach that is similar to that proposed by Berlekamp [1] to factorize a Hankel matrix which is not strongly nonsingular. Also, there is much recent interest, e.g., [3], [5], [9], [10], [12], in exploring the connections between a modified asymmetric Lanczos algorithm and orthogonal polynomials with respect to an indefinite inner product.

7 Acknowledgements

The authors acknowledge their research sponsors: D. L. Boley was supported in part by the National Science Foundation under grant CCR-8813493 and by the Minnesota Supercomputer Institute, T. J. Lee by an Intel Foundation Graduate Fellowship, and F. T. Luk by the Army Research Office under contract DAAL03-90-G-0104. The authors also thank Adam Bojanczyk and Douglas Sweet for valuable discussions.

8 References

[1] E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York, NY, 1968.

[2] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Welsey, Reading, Mass., 1983.

[3] D. L. Boley, "Error correction, orthogonal polynomials, Hankel matrices and the Lanczos algorithm," Report TR 91-8, Computer Science Dept., Univ. of Minnesota, Twin Cities, Minnesota, April 1991.

[4] D. L. Boley, R. P. Brent, G. H. Golub and F. T. Luk, "Algorithmic fault tolerance using the Lanczos method," *SIAM J. Matrix Anal.*, vol. 13 (1992), to appear.

[5] D. L. Boley, S. Elhay, G. H. Golub and M. H. Gutknecht, "Nonsymmetric Lanczos and finding orthogonal polynomials associated with indefinite weights," *Numerical Algorithms*, vol. 1 (1991), pp. 21-44.

[6] D. L. Boley and G. H. Golub, "The Lanczos algorithm and controllability," *Systems and Control Letters*, vol. 4 no. 6 (1984), pp. 317-324.

[7] D. L. Boley and G. H. Golub, "The nonsymmetric Lanczos algorithm and controllability," *Systems and Control Letters*, vol. 16 (1991) p 97-105.

[8] P. L. Chebyshev, "Sur l'interpolation par la méthode des moindres carrés," *Mém. Acad. Impér. Sci., St. Pétersbourg* (7) 1 #15 (1859), pp. 1-24 [*Oeuvres I*, pp. 473-498].

[9] R. W. Freund, M. H. Gutknecht and N. M. Nachtigal, "An implementation of the look-ahead Lanczos Algorithm for non-Hermitian matrices, part I," Numerical Analysis Report TR 90-10, Mathematics Dept., MIT, Cambridge, Mass., 1990.

[10] R. W. Freund and N. M. Nachtigal, "An implementation of the look-ahead Lanczos Algorithm for non-Hermitian matrices, part II," Numerical Analysis Report TR 90-11, Mathematics Dept., MIT, Cambridge, Mass., 1990.

[11] W. Gautschi, "On generating orthogonal polynomials," *SIAM J. Sci. Statist. Comput.*, vol. 3 (1982), pp. 289-317.

[12] G. H. Golub and M. H. Gutknecht, "Modified moments for indefinite weight functions," *Numer. Math.*, vol. 57 (1990), pp. 607-624.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd Ed., The Johns Hopkins University Press, Baltimore, Maryland, 1989.

[14] W. B. Gragg and A. Lindquist, "On the partial realization problem," *Linear Alg. Applics.*, vol. 50 (1973), pp. 277-319.

[15] M. H. Gutknecht, "A completed theory for the Lanczos algorithm," Preprint, 1989.

[16] M. H. Gutknecht, "The unsymmetric Lanczos algorithms and their relations to Padé approximation, continued fractions, the QD algorithm, biconjugate gradient squared algorithms, and fast Hankel solvers," preprint, 1990.

[17] G. Heinig and P. Jankowski, "Parallel and Superfast algorithms for Hankel systems of equations," *Numerische Mathematik*, vol. 58 (1990), pp. 109-127.

[18] A. S. Householder, *The Theory of Matrices in Numerical Analysis*, Dover, New York, New York, 1975.

[19] E. Jonckheere and C. Ma, "A simple Hankel interpretation of the Berlekamp-Massey algorithm," *Linear Alg. Applics.*, vol. 125 (1989), pp. 65-76.

[20] S. Y. Kung, "Multivariable and Multidimensional Systems: Analysis and Design," Ph.D. Dissertation, Dept. of Electrical Engineering, Stanford Univ., CA, 1977.

[21] G. Labahn, D. K. Choi and S. Cabay, "The inverses of block Hankel and block Toeplitz matrices," *SIAM J. Comput.*, vol. 19 (1990), pp. 99-123.

[22] C. Lanczos, "An iteration method for the solution of the eigenvalue problem linear differential and integral operators," *J. Res. Natl. Bur. Stand.*, vol. 45 (1950), pp. 255-282.

[23] L. Ljung, *System Identification: Theory for the User*, Prentice Hall, Englewood Cliffs, New Jersey, 1987.

[24] J. L. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15 (1967), pp. 122-127.

[25] D. Pal, "Fast triangular factorization of matrices with arbitrary rank profile," Ph.D. Dissertation, Dept. of Electrical Engineering, Stanford Univ., CA, 1990.

[26] B. N. Parlett, "Reduction to tridiagonal form and minimal realizations," Preprint, 1990.

[27] B. N. Parlett, D. R. Taylor and Z. A. Liu, "A look-ahead Lanczos algorithm for unsymmetric matrices," *Math. Comp.*, vol. 44 (1985), pp. 105-124.

[28] J. L. Phillips, "The triangular decomposition of Hankel matrices," *Math. Comp.*, vol. 25, No. 115 (1971), pp. 599-602.

[29] J. Rissanen, "Algorithms for triangular decomposition of block Hankel and Toeplitz matrices with application to factoring positive matrix polynomials," *Math. Comp.*, vol. 27, No. 121 (1973), pp. 147-154.

[30] J. Rissanen, "Solution of linear equation with Hankel and Toeplitz matrices," *Numer. Math.*, vol. 22 (1974), pp. 361-366.

[31] W. F. Trench, "An algorithm for the inversion of finite Hankel matrices," *J. Soc. Indust. Appl. Math.*, vol. 13 (1965), pp. 1102-1107.

[32] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.