JOURNAL OF COMPLEXITY 10, 142–164 (1994)

A Multi-step Algorithm for Hankel Matrices

ADAM W. BOJANCZYK

Cornell University, School of Electrical Engineering, Ithaca, New York 14853-3801

AND

GEORG HEINIG

Universität Leipzig, Fachbereich Mathematik/Informatik, D-O-7010 Leipzig, Germany

Received September 30, 1992

DEDICATED TO JOSEPH F. TRAUB ON THE OCCASION OF HIS 60TH BIRTHDAY

In this paper we discuss a Schur-type algorithm for solving indefinite Hankel systems of linear equations. The algorithm is derived using the language of linear algebra. © 1994 Academic Press, Inc.

1. INTRODUCTION

Algorithms for inversion and triangularization of strongly nonsingular Hankel matrices were developed by Berlekamp (1968) and Massey (1967), Phillips (1971), Rissanen (1973), Trench (1965), and others. Generalizations to Hankel matrices with singular principal minors (singular sections) were considered by Heinig (1984), Delsarte *et al.* (1985), Pal and Kailath (1990), Rissanen (1973), Zarowski (1991), and others. However, these algorithms may produce very inaccurate numerical results even if the matrix is well-conditioned and strongly nonsingular. Hence they require exact arithmetics in order to obtain reliable results. The reason for the instabilities can be explained as follows. All the algorithms compute certain quantities corresponding to the principal submatrices of the matrix recursively, and ill-conditioned sections produce an instable evolution of roundoff errors.

0885-064X/94 $6.00

Recently there have been many attempts to stabilize algorithms for solving systems of Hankel and Toeplitz linear equations (Bojanczyk *et al.*, 1991; Chan and Hanson, 1990; Cabay and Meleshko, 1991; Freund and Zha, 1992; Gukknecht, 1992). All approaches presented (so far) in the literature are based on a "look-ahead" strategy, which is common in other fields of numerical linear algebra, especially in Lanczos methods for nonsymmetric matrices. "Look-ahead strategy" means to jump from one well-conditioned section to the next one. In the case in which one has only a few ill-conditioned sections one will obtain in this way an algorithm which is fast and (weakly) stable. It is remarkable that the different authors use different languages for the derivations of their algorithms. In Cabay and Meleshko (1991), and Gutknecht (1992) the relation of Hankel and Toeplitz matrices to the Padé approximation problem is utilized and the language is "polynomial." In Freund and Zha (1992) formally orthogonal polynomials play a central role. Only in Chan and Hanson (1990) is the derivation carried out in pure matrix language.

The motivation for this paper came from the paper by Chan and Hanson (1990), which was probably the first attempt to produce a stable algorithm for nonsymmetric structured systems of equations, and in which an interesting look-ahead approach to stabilizing fast solvers for the case of Toeplitz matrices was presented. The algorithm offered can be considered as a (weakly stable) generalization of the famous Levinson algorithm. In this paper we apply the idea of Chan and Hanson in order to get fast and weakly stable Hankel solvers. However, our algorithm is a Schur-type algorithm rather then a Levinson-type algorithm. Recall that the Schur algorithm computes the factors of the LDU decomposition of the matrix, whereas the Levinson algorithm finds the factors of the inverse matrix.

Our approach is based on the algebraic properties of Schur complements of Hankel matrices. These properties are presented in Section 3. In Section 2, which has preliminary character, we collect some well-known material concerning inversion and factorization of Hankel matrices. In Section 4 the look-ahead (multi-step) algorithm is described.

Let us remark that in Heinig (1992) a different (not look-ahead) approach to obtaining stable Hankel and Toeplitz solvers is described. The approach is based on the idea of transforming Hankel or Toeplitz systems into systems the coefficient matrix of which is a generalized Cauchy matrix (or Loewner matrix). For such systems pivoting can be applied (however, extensive pivoting may slow down the algorithm).

Throughout this paper we use a MATLAB-type convention for denoting submatrices. If A is an $m \times n$ matrix then $A(i : j, k : l)$ denotes a submatrix of A obtained by removing from A rows 0 to $i - 1$ and $j + 1$ to $m - 1$, and columns 0 to $k - 1$ and $l + 1$ to $n - 1$. Also, by the MATLAB convention, $A(i : j, k) = A(i : j, k : k)$ and $A(:, k : l) = A(0 : m - 1, k : l)$. In

formulas we sometimes use an alternative notation for $A(i : j, k : l)$, namely $A_{i:j,k:l}$.

Let e_i be a vector of length n with the ith component equal to 1 and 0 otherwise. We use the following special matrices:

$$J_n \equiv \sum_{i=1}^{n} e_i e_{n+1-i}^T = \begin{pmatrix} 0 & & & 1 \\ & & \cdot & \\ & \cdot & & \\ 1 & & & 0 \end{pmatrix}, \tag{1.1a}$$

$$Z_n \equiv \sum_{i=1}^{n-1} e_{i+1} e_i^T = \begin{pmatrix} 0 & & \cdots & \cdots & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}. \tag{1.1b}$$

The matrix J_n is the *reverse* ordering while Z_n is the *shift-down* operator in C^n. Finally, if $z = (\zeta_0, \ldots, \zeta_{n-1})^T$ is a vector of dimension n then we define

$$|z| \stackrel{\text{def}}{=} (|\zeta_0|, \ldots, |\zeta_{n-1}|)^T.$$

2. Inversion and Triangularization

In this section we recall some well-known facts concerning the inversion and triangular factorization of a nonsingular Hankel matrix.

Let $H = (\eta_{i+j})_{i,j=0}^{N-1}$ denote a nonsingular $N \times N$ Hankel matrix, and $g = [\eta_N, \eta_{N+1}, \ldots, \eta_{2N-1}]^T$ be an N-dimensional vector (η_{2N-1} is an arbitrary scalar). That is,

$$[H\ g] = \begin{pmatrix} \eta_0 & \eta_1 & \cdots & \eta_{N-1} & \eta_N \\ \eta_1 & \eta_2 & \cdots & \eta_N & \eta_{N+1} \\ \vdots & \vdots & \vdots & \vdots & \\ \eta_{N-1} & \eta_N & \cdots & \eta_{2N-2} & \eta_{2N-1} \end{pmatrix}.$$

Together with H and g we find it convenient to consider an extended $(2N-1) \times N$ Hankel matrix $\hat{H} = (\eta_{i+j})_{i=0,\ldots,2N-2, j=0,\ldots,N-1}$, and an extended vector $\hat{g} = (\eta_i)_{i=N,\ldots,3N-2}$,

$$[\hat{H}\ \hat{g}] = \begin{pmatrix} \eta_0 & \eta_1 & \cdots & \eta_{N-1} & \eta_N \\ \eta_1 & \eta_2 & \cdots & \eta_N & \eta_{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \eta_{N-1} & \eta_N & \cdots & \eta_{2N-2} & \eta_{2N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \eta_{2N-2} & \eta_{2N-1} & \cdots & \eta_{3N-3} & \eta_{3N-2} \end{pmatrix}, \tag{2.1}$$

where η_j, $j = 2N, \ldots, 3N-2$, are arbitrary scalars.

By rearranging the columns or rows, the Hankel matrix can be transformed into a Toeplitz matrix. For example, we can reorder the columns of H from the last to the first and obtain a Toeplitz matrix:

$$T = (\eta_{N+i-j})_{i,j=0}^{N-1} = HJ_N = \begin{pmatrix} \eta_{N-1} & \eta_{N-2} & \cdots & \eta_0 \\ \eta_N & \eta_{N-1} & \cdots & \eta_1 \\ \vdots & \vdots & \vdots & \vdots \\ \eta_{2N-2} & \eta_{2N-3} & \cdots & \eta_{N-1} \end{pmatrix}. \tag{2.2}$$

Thus an Toeplitz linear solver like the Levinson (1946) or Bareiss (1969) algorithms can be applied to Hankel linear systems. However, in some applications it is necessary to solve intermediate Hankel linear systems defined by the principal submatrices of H and hence algorithms dealing with the Hankel structure directly are of interest.

The inverse of H is determined by the solutions of the two equations

$$Hx = e_N \qquad Hy = g, \tag{2.3}$$

where e_N is the unit vector of the Nth coordinate in C^N. We call x and y the *fundamental* solutions. Furthermore, the solvability of Eq. (2.3) implies the nonsingularity of H. When $x = [\xi_0, \xi_1, \ldots, \xi_{N-1}]^T$ and $y = [\psi_0, \psi_1, \ldots, \psi_{N-1}]^T$ are known, one of the possible ways to represent the inverse H^{-1} is the Gohberg–Semencul-type formula

$$H^{-1} = \begin{pmatrix} \xi_0 & & & \\ \xi_1 & \xi_0 & & \\ \vdots & \vdots & \ddots & \\ \xi_{N-1} & \xi_{N-2} & \cdots & \xi_0 \end{pmatrix} \begin{pmatrix} -\psi_1 & \cdots & -\psi_{N-1} & 1 \\ \vdots & & \iddots & \\ -\psi_{N-1} & \iddots & & \\ 1 & & & \end{pmatrix} + \begin{pmatrix} \psi_0 & & & 0 \\ \psi_1 & \psi_0 & & \\ \vdots & \vdots & \ddots & 0 \\ \psi_{N-1} & \psi_{N-2} & \cdots & \psi_0 \end{pmatrix} \begin{pmatrix} \xi_0 & \cdots & \xi_{N-1} & 0 \\ \vdots & & \iddots & \\ \xi_{N-1} & \iddots & & \\ 0 & & & \end{pmatrix}. \tag{2.4}$$

The importance of formula (2.4) lies in the fact that it allows us to solve a Hankel system of linear equations in $O(N \log N)$ operations if x and y are given.

Together with H we consider the family of nested submatrices $H_n = H(0 : n-1, 0 : n-1) = (\eta_{i+j})_{i,j=0}^{n-1}$, vectors $g^{(n)} = H(0 : n-1, n) = [\eta_n, \ldots, \eta_{2n-1}]^T$, and the corresponding fundamental solutions $x^{(n)}$ and $y^{(n)}$. Associated with the pair of the fundamental solutions $\{x^{(n)}\ y^{(n)}\}$ is a pair of *residual* vectors $\hat{p}^{(n)} = (\pi_i^{(n)})_{i=0}^{2N-2}$ $\hat{r}^{(n)} = (\rho_i^{(n)})_{i=0}^{2N-2}$ defined as

$$\hat{p}^{(n)} \equiv \hat{H}(:, 0 : n-1)x^{(n)} \qquad \text{and} \qquad \hat{r}^{(n)} \equiv \hat{H}(:, 0 : n) \begin{pmatrix} -y^{(n)} \\ 1 \end{pmatrix}, \tag{2.5}$$

or componentwise

$$\pi_i^{(n)} \equiv [\eta_i, \ldots, \eta_{i+n-1}]x^{(n)}, \quad \text{and} \quad \rho_i^{(n)} \equiv [\eta_i, \ldots, \eta_{i+n}] \begin{pmatrix} -y^{(n)} \\ 1 \end{pmatrix}, \tag{2.6}$$

for $i = 0, \ldots, 2N-2$ and $n = 0, \ldots, N-1$. Note that from the definition (2.3), $\rho_i^{(n)} = 0$ for $i < n$, $\pi_i^{(n)} = 0$ for $i < n-1$ and $\pi_{n-1}^{(n)} = 1$.

The Berlekamp–Massey and related algorithms construct vectors $\{x^{(n)}\ y^{(n)}\}$ and $\{\hat{p}^{(n)}\ \hat{r}^{(n)}\}$ recursively. The desired solutions of (2.3) are $x = x^{(N-1)}$ and $y = y^{(N-1)}$). A possible way of deriving a recursive relationship between $\{x^{(n)}\ y^{(n)}\}$ and $\{x^{(n+1)}\ y^{(n+1)}\}$ is by considering the following $(n+1)$-dimensional vector,

$$\begin{pmatrix} 0 \\ y^{(n)} \end{pmatrix}, \begin{pmatrix} -y^{(n)} \\ 1 \end{pmatrix}, \begin{pmatrix} x^{(n)} \\ 0 \end{pmatrix}.$$

Note that if H_n is nonsingular then

$$H_{n+1}\begin{pmatrix} 0 & -y^{(n)} & x^{(n)} \\ y^{(n)} & 1 & 0 \end{pmatrix} = \begin{pmatrix} \eta_{n+1} & 0 & 0 \\ \vdots & \vdots & \vdots \\ \eta_{2n-1} & 0 & 0 \\ \eta_{2n} - \rho_n^{(n)} & 0 & 1 \\ \eta_{2n+1} - \rho_{n+1}^{(n)} & \rho_n^{(n)} & \pi_n^{(n)} \end{pmatrix}. \tag{2.7}$$

If H_{n+1} is also nonsingular then $\rho_n^{(n)} \neq 0$. Now by postmultiplying (2.7) by a 3×3 elimination matrix W_n,

$$W_n = \begin{pmatrix} 1 & 0 & 0 \\ w_{2,1}^{(n)} & w_{2,2}^{(n)} & 0 \\ w_{3,1}^{(n)} & 0 & 1 \end{pmatrix},$$

where

$$w_{2,1}^{(n)} = \frac{\rho_{n+1}^{(n)} - \rho_n^{(n)}\pi_n^{(n)}}{\rho_n^{(n)}}, \quad w_{3,1}^{(n)} = \rho_n^{(n)}, \quad w_{2,2}^{(n)} = \frac{1}{\rho_n^{(n)}},$$

we obtain

$$\begin{pmatrix} \eta_{n+1} & 0 & 0 \\ \vdots & \vdots & \vdots \\ \eta_{2n-1} & 0 & 0 \\ \eta_{2n} - \rho_n^{(n)} & 0 & 1 \\ \eta_{2n+1} - \rho_{n+1}^{(n)} & \rho_n^{(n)} & \pi_n^{(n)} \end{pmatrix} W_n = \begin{pmatrix} \eta_{n+1} & 0 & 0 \\ \vdots & \vdots & \vdots \\ \eta_{2n-1} & 0 & 0 \\ \eta_{2n} & 0 & 1 \\ \eta_{2n+1} & 1 & \pi_n^{(n)} \end{pmatrix}. \tag{2.8}$$

Thus the following relation holds true:

$$\begin{pmatrix} 0 & -y^{(n)} & x^{(n)} \\ y^{(n)} & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} W_n = \left(y^{(n+1)}, \quad x^{(n+1)}, \quad \begin{pmatrix} x^{(n)} \\ 0 \end{pmatrix} \right).$$

In other words,

$$x^{(n+1)} = \frac{1}{\rho_n^{(n)}} \begin{pmatrix} -y^{(n)} \\ 1 \end{pmatrix}, \tag{2.9}$$

$$y^{(n+1)} = \begin{pmatrix} 0 \\ y^{(n)} \end{pmatrix} + \frac{\rho_{n+1}^{(n)} - \rho_n^{(n)}\pi_n^{(n)}}{\rho_n^{(n)}} \begin{pmatrix} -y^{(n)} \\ 1 \end{pmatrix} + \rho_n^{(n)} \begin{pmatrix} x^{(n)} \\ 0 \end{pmatrix}. \tag{2.10}$$

Remark 1: Note that in the case when H_{n-1} is also nonsingular, $x^{(n)}$ can be replaced by $y^{(n-1)}$. Indeed, according to (2.9) we have

$$x^{(n)} = \frac{1}{\rho_{n-1}^{(n-1)}} \begin{pmatrix} -y^{(n-1)} \\ 1 \end{pmatrix},$$

which implies

$$\pi_i^{(n)} = \frac{1}{\rho_{n-1}^{(n-1)}} \rho_i^{(n-1)}.$$

Thus (2.10) goes over into

$$\begin{pmatrix} -y^{(n+1)} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -y^{(n)} \\ 1 \end{pmatrix} + \left(\frac{\rho_n^{(n-1)}}{\rho_{n-1}^{(n-1)}} - \frac{\rho_{n+1}^{(n)}}{\rho_n^{(n)}} \right) \begin{pmatrix} -y^{(n)} \\ 1 \\ 0 \end{pmatrix} + \frac{\rho_n^{(n)}}{\rho_{n-1}^{(n-1)}} \begin{pmatrix} -y^{(n-1)} \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{2.11}$$

The relation (2.11) is the familiar three-term recurrence formula which shows a connection between the Berlekamp–Massey algorithm, the Lanczos-type computation, and the Chebyshev algorithm for generating orthogonal polynomials from their moments (Householder, 1964; Kung, 1977).

From the relations (2.9) and (2.10) it is not difficult to derive the following recursions for the vectors $\hat{r}^{(n)}$ and $\hat{p}^{(n)}$, $n = 0, \ldots, N-1$,

$$\hat{r}^{(n)} = Z_{2N}^T r^{(n)} + \frac{\rho_{n+1}^{(n)} - \rho_n^{(n)}\pi_n^{(n)}}{\rho_n^{(n)}} r^{(n)} - \rho_n^{(n)} p^{(n)}, \tag{2.12}$$

$$\hat{p}^{(n+1)} = \frac{1}{\rho_n^{(n)}} \hat{r}^{(n)}. \tag{2.13}$$

The relations (2.9) and (2.10) together with (2.6) completely define the recurrence for computing the fundamental solutions x and y. The computation of the elements of the residuals via formula (2.6) requires evaluations of scalar products. This can be avoided by making use of the recurrences (2.12) and (2.13) which are more amenable for parallel computing.

It is not difficult to see that the cost of either of these procedures is $O(N^2)$ arithmetic operations for an $n \times n$ Hankel matrix H.

The vectors $y^{(n)}$ and $\hat{r}^{(n)}$ are closely related to the triangular LDU factorization of H and H^{-1}. Suppose that H is strongly nonsingular (that is, all submatrices H_n are nonsingular). Let

$$u^{(n)} = \begin{pmatrix} y^{(n)} \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

be an N-dimensional vector. Then

$$U = [u^{(0)}, u^{(1)}, \ldots, u^{(N-1)}]$$

is the upper triangular factor of the LDU decomposition of N^{-1}. Furthermore, let

$$(l^{(n)})^T = \frac{1}{\rho_n^{(n)}} [0, \ldots, 0, \rho_n^{(n)}, \ldots, \rho_{N-1}^{(n)}]$$

be an N-dimensional vector. Then

$$L = [l^{(0)}, l^{(1)}, \ldots, l^{(N-1)}]$$

is the lower triangular factor of the LDU decomposition of H. The diagonal factor is given by

$$D = \operatorname{diag}(\rho_0^{(0)}, \rho_1^{(1)}, \ldots, \rho_{N-1}^{(N-1)}).$$

Since H is symmetric, the other factors are obtained by transposition; that is, we have

$$H^{-1} = U^T D U \quad \text{and} \quad H = L D L^T.$$

The quantities $\rho_n^{(n)}$ play a decisive role for the recursive algorithms. If $\rho_n^{(n)} = 0$ then H_{n+1} is singular and the algorithm breaks down. Conversely, the singularity of H_{n+1} implies $\rho_n^{(n)} = 0$. Furthermore, a small in the magnitude $\rho_n^{(n)}$ indicates that H_{n+1} is ill-conditioned. This can be seen from the formula

$$H_{n+1}^{-1} = \begin{pmatrix} H_n^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{\rho_n^{(n)}} \begin{pmatrix} -y^{(n)} \\ 1 \end{pmatrix} (-y^{(n)} \quad 1), \tag{2.14}$$

which is consequence of the Schur complement formula (3.2). From this we get the estimation

$$\|H_{n+1}^{-1}\| \leq \|H_n^{-1}\| + \frac{\tau_n^2}{|\rho_n^{(n)}|}, \tag{2.15}$$

where

$$\tau_n = \left\| \begin{pmatrix} -y^{(n)} \\ 1 \end{pmatrix} \right\| (\geq 1).$$

This estimation shows that in order to judge about the condition number of H_{n+1}, one has also to take into account the magnitude of τ_n. The following example illustrates the case.

EXAMPLE 1. Suppose that $n = 2$ and

$$H_{n+1} = \begin{pmatrix} -\gamma & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1+\gamma \end{pmatrix},$$

where γ is of "moderate" size and γ^2 is "large." We have

$$H_n^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & \gamma \end{pmatrix}, \quad y^{(n)} = \begin{pmatrix} 1 \\ \gamma \end{pmatrix}, \quad \rho_n^{(n)} = 1.$$

The magnitude of $\rho_n^{(n)}$ seems to indicate that H_{n+1} is well-conditioned, but the (1,1)-element[1] of H_{n+1}^{-1} equals $\gamma^2 + \gamma$. Hence H_{n+1} is ill-conditioned.

The example is also instructive in another context. It shows that for the construction of stable algorithms it does not suffice to use the quantities $\rho_n^{(n)}$ for step size regulation; it also shows that, unlike for positive definite Toeplitz matrices (see [6]), the condition of the upper and lower triangular factors of the LDU decomposition is not determined by the diagonal factor.

In constructing the vectors x and y, in order to avoid problems with singular and ill-conditioned submatrices H_n, one can move from one well-

[1] We enumerate the entries of an $n \times n$ matrix from 0 to $n - 1$.

conditioned submatrix to the next one. This type of procedure, which is known as a *look-ahead* procedure, was first proposed in Chan and Hansen (1990) for dealing with near singularities in the Levinson algorithm for solving indefinite systems of Toeplitz equations. In this paper we consider an analogous approach for solving indefinite Hankel systems of linear equations. A basis for a such multi-step algorithm is derived and explained in the subsequent sections.

3. Algebraic Properties of Schur Complement

As a preliminary step for the construction of a multi-step algorithm we study the algebraic properties of Schur complements for Hankel matrices.

Let A be a matrix partitioned in the form

$$A = \begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix}.$$

If A_0 is nonsingular then

$$\begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix} \begin{pmatrix} I & A_0^{-1}A_1 \\ 0 & I \end{pmatrix} = \begin{pmatrix} A_0 & 0 \\ A_2 & A_3 - A_2A_0^{-1}A_1 \end{pmatrix}.$$

The matrix $S \equiv A_3 - A_2A_0^{-1}A_1$ is called the *Schur complement of* A_0 with respect to A. From this definition it is easily seen that for $Y = A_0^{-1}$ and $W = A_2A_0^{-1}$,

$$A = \begin{pmatrix} I_0 & 0 \\ W & I_3 \end{pmatrix} \begin{pmatrix} A_0 & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I_0 & Y \\ 0 & I_3 \end{pmatrix}, \tag{3.1}$$

where I_0 and I_3 denote unit matrices of appropriate sizes, holds. The relation (3.1) implies

$$A^{-1} = \begin{pmatrix} A_0^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -Y \\ I_3 \end{pmatrix} S^{-1}(-W \quad I_3) \tag{3.2}$$

which gives a formula for the inverse of A when inverses of A_0 and the Schur complement S are known. The formula (3.2) can be used to compute the solution to a linear system

$$\begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix} z = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}$$

when the solution z_0 to $A_0 z_0 = b_0$ is known. Indeed, the solution z_0 can be updated to the solution z as follows

$$z = \begin{pmatrix} z_0 \\ 0 \end{pmatrix} + \begin{pmatrix} -Y \\ I_3 \end{pmatrix} S^{-1}(b_1 - A_2 z_0). \tag{3.3}$$

We specify now formulas (3.1) and (3.2) for Hankel matrices. In the remainder of the section let n be fixed. So we may omit the superscript "n" for all quantities. Let $H_n = H(0 : n-1, 0 : n-1)$ be nonsingular and define $g_i = H(0 : n-1, n+i)$ for $i = 0, \ldots, N-n-1$. Furthermore, let $H_{n+k} = H(0 : n+k-1, 0 : n+k-1)$ be partitioned as

$$H_{n+k} = \begin{pmatrix} H_n & G_k \\ G_k^T & F_k \end{pmatrix},$$

where $G_k = (g_0, \ldots, g_{k-1}) = H(0 : n-1, n : n+k-1)$ and $F_k = H(n : n+k-1, n : n+k-1)$. Define now an $n \times k$ matrix Y_k,

$$Y_k \equiv G_n^{-1} G_k = (y_0, \ldots, y_{k-1}). \tag{3.4}$$

The equations $H_n y_i = g_i$ are called *generalized Yule–Walker equations* and the vectors $y_i = H_n^{-1} g_i$, $i = 0, \ldots, N-n-1$, are their solutions. The case $i = 0$ is the case of the classical Yule–Walker equation. The Schur complement $S_k = (\sigma_{ij})_{i,j=0}^{k-1}$ of H_n with respect to H_{n+k} is then defined as follows

$$S_k = (G_k^T \quad F_k) \begin{pmatrix} -Y_k \\ I_k \end{pmatrix} = (s_0, \ldots, s_{k-1}).$$

The importance of the Schur complement is in its later use in the determination of the fundamental solutions corresponding to H_{n+k}.

It is convenient to consider together with S_k the extended matrix $\hat{S}_k$,

$$\hat{S}_k = \hat{H}(:, 0 : k-1) \begin{pmatrix} -Y_k \\ I_k \end{pmatrix}.$$

The columns of $\hat{S}_k$ are residual vectors corresponding to the vectors $y_0, \ldots, y_{k-1}$. Thus $S_k = \hat{S}_k(n : n+k-1, 0 : k-1)$, and also $\hat{S}_k(0 : n-1, 0 : k-1) = 0$.

Our aim is now to construct recursions for the columns of Y_k and $\hat{S}_k$. For this we use the vectors $x \equiv H_n^{-1} e_n$ and $y_0 \equiv H_n^{-1} g_0$ and the corresponding residual vectors $\hat{p}$ and $\hat{r}$ defined by

$$\hat{p} \equiv \hat{H}(:, 0 : n-1)x \qquad \text{and} \qquad \hat{r} \equiv \hat{H}(:, 0 : n)\begin{pmatrix} -y_0 \\ 1 \end{pmatrix}. \tag{3.5}$$

Furthermore we utilize the following relations

$$H_n Z - Z^T H_n = e_n g_0^T - g_0 e_n^T, \tag{3.6}$$

$$g_{j+1} = Z^T g_j + \eta_{2n+j} e_n, \tag{3.7}$$

$$\pi_{i+n} = g_i^T x = y_i^T H_n x = e_n^T y_i, \tag{3.8}$$

$$\rho_{i+n} = (g_i^T \quad \eta_{2n+i})\begin{pmatrix} -y_0 \\ 1 \end{pmatrix} = \eta_{2n+i} - g_0^T y_i. \tag{3.9}$$

THEOREM 1. *Let* $g_i = H(0 : n-1, n+i)$, $i = 0, 1, \ldots, N-n-1$. *The vectors* $y_i = H_n^{-1} g_i$ *satisfy the recursion*

$$y_{j+1} = Z y_j + \rho_{n+j} x + \pi_{n+j} y_0. \tag{3.10}$$

Moreover, the matrix Y_k *can be computed by the global formula*

$$Y_k = \begin{pmatrix} \xi_0 & & & \\ \xi_1 & \xi_0 & & \\ \vdots & \vdots & \ddots & \\ \xi_{k-1} & \xi_{k-2} & \cdots & \xi_0 \\ \vdots & \vdots & & \vdots \\ \xi_{n-1} & \xi_{n-2} & \cdots & \xi_{n-k} \end{pmatrix} \begin{pmatrix} 0 & \rho_n & \rho_{n-1} & \cdots & \rho_{n+k-2} \\ & 0 & \rho_n & \cdots & \rho_{n+k-3} \\ & & \ddots & \ddots & \vdots \\ & & & 0 & \rho_n \\ & & & & 0 \end{pmatrix}$$

$$+ \begin{pmatrix} \psi_{0,0} & & & \\ \psi_{0,1} & \psi_{0,0} & & \\ \vdots & \vdots & \ddots & \\ \psi_{0,k-1} & \psi_{0,k-2} & \cdots & \psi_{0,0} \\ \vdots & \vdots & & \vdots \\ \psi_{0,n-1} & \psi_{0,n-2} & \cdots & \psi_{0,n-k} \end{pmatrix} \begin{pmatrix} 1 & \pi_n & \pi_{n-1} & \cdots & \pi_{n+k-2} \\ & 1 & \pi_n & \cdots & \pi_{n+k-3} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & \pi_n \\ & & & & 1 \end{pmatrix}, \tag{3.11}$$

where $x = (\xi_i)_{i=0}^{n-1}$ *and* $y_0 = (\psi_{0,i})_{i=0}^{n-1}$.

Proof. From (3.6) we get

$$Z^T g_j = Z^T H_n y_j = H_n Z y_j - e_n g_0^T y_j + g_0 e_n^T y_j$$

and from (3.7)

$$Z^T g_j = g_{j+1} - \eta_{2n+j} e_n.$$

Comparing these relations and applying H_n^{-1} we obtain

$$y_{j+1} - \eta_{2n+j} x = Z y_j - x g_0^T y_j + y_0 e_n^T y_j.$$

Taking into account (3.8) and (3.9) we get

$$y_{j+1} - \eta_{2n+j} x = Z y_j + (\rho_{j+n} - h_{2n+j}) x + \pi_{j+n} y_0,$$

which is equivalent with (3.10). Applying now (3.10) successively we obtain

$$y_j = \rho_{j+n-1} x + \rho_{i+n-2} Z x + \cdots + \rho_n Z^{j-1} x + \pi_{j+n-1} y_0 + \cdots + \pi_{n-1} Z^j y_0,$$

which is just another form of (3.11). ∎

Now we show how to compute the Schur complement S_k. The following remark is important.

Remark. If $l > k$ then $\hat{S}_k$ is a principal submatrix of $\hat{S}_l$.

Due to this remark it suffices to compute $\hat{S}$ where

$$\hat{S} \equiv \hat{S}_{N-n} = (\hat{s}_0 \cdots \hat{s}_{N-n-1}) = (\sigma_{i,j})_{i=0}^{2N-1}{}_{j=0}^{N-n-1}.$$

THEOREM 2. *The columns $\hat{s}_i$ and $\hat{S}$ satisfy the recursion*

$$\hat{s}_0 = \hat{r},\ \hat{s}_{j+1} = Z^T \hat{s}_j - \rho_{n+j} \hat{p} + \pi_{n+j} \hat{r}, \tag{3.12}$$

where $\hat{r} = (\pi_j)_{j=0}^{2N-1}$ and $\hat{r} = (\rho_j)_{j=0}^{2N-1}$ are the residual vectors corresponding to the fundamental solutions x and y. The matrix $S = \hat{S}(n : N-1, 0 : N-n-1)$ can be expressed by the formula

$$S = -\begin{pmatrix} \pi_n & \cdots & \cdots & \pi_{N-1} \\ \pi_{n+1} & \cdots & \pi_{N-1} & \\ \vdots & \iddots & & \\ \pi_{N-1} & & & \end{pmatrix} \begin{pmatrix} 0 & \rho_n & \rho_{n-1} & \cdots & \rho_{N-2} \\ & 0 & \rho_n & \cdots & \rho_{N-3} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & \rho_n \\ & & & & 0 \end{pmatrix}$$

$$+\begin{pmatrix} \rho_n & \cdots & \cdots & \rho_{N-1} \\ \rho_{n+1} & \cdots & \rho_{N-1} & \\ \vdots & \iddots & & \\ \rho_{N-1} & & & \end{pmatrix} \begin{pmatrix} 1 & \pi_n & \pi_{n-1} & \cdots & \pi_{N-2} \\ & 1 & \pi_n & \cdots & \pi_{N-3} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & \pi_n \\ & & & & 1 \end{pmatrix}. \tag{3.13}$$

Proof. From (3.10) we conclude

$$g_i^T y_{j+1} = g_i^T Z y_j + \rho_{n+j} g_i^T x + \pi_{n+j} g_i^T y_0.$$

Taking (3.7) into account we obtain

$$g_i^T y_{j+1} = (g_{i+1}^T - \eta_{2n+i} e_n^T) y_j + \rho_{n+j} g_j^T x + \pi_{n+j} g_j^T y_0.$$

Hence

$$\sigma_{i+1,j} - \sigma_{i,j+1} = g_i^T y_{j+1} - g_{i+1}^T y_j = \pi_i \rho_{n+j} - \rho_i \pi_{n+j}.$$

This is equivalent to (3.12). Applying now (3.12) successively we get

$$\sigma_j = -\rho_{n+j-1} p - \cdots - \rho_n (Z^T)^{j-1} p + \pi_{n+j-1} r + \cdots + \pi_n (Z^T)^{j-1} r + (Z^T)^j r,$$

which is another form of (3.13). ∎

Note that in particular

$$S_1 = \rho_n \qquad \text{and} \qquad S_2 = \begin{pmatrix} \rho_n & \rho_{n+1} \\ \rho_{n+1} & \rho_{n+2} \end{pmatrix}.$$

Hence, in this case we have

$$\|S_1^{-1}\| = |\rho_n|^{-1} \qquad \text{and} \qquad \|S_2^{-1}\| = |\rho_n \rho_{n+2} - \rho_{n+1}^2|^{-1} \|S_2\|.$$

We remark a slight modification of the formula (3.13).

COROLLARY. The matrix S_k can be represented as

$$S_k = \begin{pmatrix} 0 & \cdots & \cdots & \sigma_0 \\ \vdots & & \iddots & \sigma_1 \\ \vdots & \iddots & \iddots & \vdots \\ \sigma_0 & \sigma_1 & \cdots & \sigma_{k-1} \end{pmatrix}$$

$$- \begin{pmatrix} \pi_n & \pi_{n+1} & & \pi_{n+k-2} & 0 \\ \pi_{n+1} & \cdots & & & \\ \vdots & & \iddots & & \\ \pi_{n+k-2} & & & & \\ 0 & & & & \end{pmatrix} \begin{pmatrix} 0 & \rho_n & \rho_{n-1} & \cdots & \rho_{n+k-2} \\ & 0 & \rho_n & \cdots & \rho_{n+k-3} \\ & & \ddots & & \vdots \\ & & & \ddots & \rho_n \\ & & & & 0 \end{pmatrix}$$

$$+ \begin{pmatrix} \rho_n & \rho_{n+1} & \cdots & \rho_{n+k-2} & 0 \\ \rho_{n+1} & & & & \\ \vdots & & & & \\ \rho_{n+k-2} & & & & \\ 0 & & & & \end{pmatrix} \begin{pmatrix} 1 & \pi_n & \pi_{n-1} & \cdots & \pi_{n+k-2} \\ & 1 & \pi_n & \cdots & \pi_{n+k-3} \\ & & \ddots & & \vdots \\ & & & \ddots & \pi_n \\ & & & & 1 \end{pmatrix}, \quad (3.14)$$

where $\sigma_j = \sigma_{n+k+j-1,0}$.

The formula (3.14) states that if $\rho_n, \ldots, \rho_{n+k-2}$ are small compared with $\rho_{n+k-1} = \sigma_{n+k-1}$, then S_k is a small perturbation of a lower triangular Hankel matrix. This observation is a basis for an *inexact* version of the Berlekamp–Messy algorithm [3] where the Schur complement S_k is replaced by a lower triangular Hankel matrix whenever $\rho_n, \ldots, \rho_{n+k-2}$ are small compared with $\rho_{n+k-1} = \sigma_{n+k-1,0}$. Such a modified algorithm produces an approximate decomposition of the original matrix H. Then the fundamental solutions x and y can be obtained by means of iterative refinement. This is at first glance very attractive. However, two things can go wrong. First, the approximation to S_k can be very ill-conditioned. Second, even if the approximation to S_k is well conditioned, the matrix H_{n+k} does not have to be; see Example 1. Thus the resulting approximate decomposition of H can have a very poor accuracy. Hence many steps of iterative refinement may be needed to obtain a good approximation of the fundamental solutions increasing the cost of such an approach to possibly unacceptable levels. In extreme cases the iterative refinement can fail to

produce any accurate digit in the computed x and y. For details see Bojanczyk *et al.* (1991).

4. Multistep Algorithm

We present now the multistep algorithm for computing the fundamental solutions x and y. First we explain the algebraic setting and after that we discuss criteria for choosing step sizes.

Suppose we are given $x^{(n)} = H^{-1}e_n$ and $y_0^{(n)} = H_n^{-1}g_0^{(n)}$ and the corresponding residual vectors $\hat{p}^{(n)}$ and $\hat{r}^{(n)}$. Further, let k be a convenient step size, i.e., an integer such that H_{n+k} is well-conditioned.

As the first step we compute the first $k+1$ columns of $\hat{S}^{(n)}$ according to Theorem 2 and the vectors $y_1^{(n)}, \ldots, y_k^{(n)}$ by Theorem 1. Since H_{n+k} is assumed to be well-conditioned the matrix $S_k^{(n)} = \hat{S}^{(n)}(n : n+k-1, 0 : k-1)$ is nonsingular.

Our aim is to compute $x^{(n+k)}$, $y_0^{(n+k)}$, $\hat{p}^{(n+k)}$, and $\hat{r}^{(n+k)}$. This can be done by the following updating formulas.

THEOREM 3. *Let* $s_k^{(n)} = S^{(n)}(n : n+k-1, k)$ *and* $Y_k^{(n)} = [y_0^{(n)}, \ldots, y_{k-1}^{(n)}]$. *Suppose that* $u_k^{(n)}$ *and* $v_k^{(n)}$ *are the solutions of the equations*

$$S_k^{(n)}u_k^{(n)} = e_k \qquad \text{and} \qquad S_k^{(n)}v_k^{(n)} = s_k^{(n)}. \tag{4.1}$$

Then

$$x^{(n+k)} = \begin{pmatrix} -Y_k^{(n)} \\ I_k \end{pmatrix} u_k^{(n)}, \tag{4.2}$$

$$y_0^{(n+k)} = \begin{pmatrix} -Y_k^{(n)} \\ I_k \end{pmatrix} v_k^{(n)} + \begin{pmatrix} y_k^{(n)} \\ 0 \end{pmatrix}, \tag{4.3}$$

$$\hat{p}^{(n+k)} = \hat{S}^{(n)} \begin{pmatrix} u_k^{(n)} \\ 0 \end{pmatrix}, \qquad \hat{r}^{(n+k)} = \hat{S}^{(n)} \begin{pmatrix} -v_k^{(n)} \\ 1 \\ 0 \end{pmatrix}, \tag{4.4}$$

where 0 *denotes a zero vector of appropriate length.*

Proof. By the definition (3.4) of $Y_k^{(n)}$ we have

$$\hat{H}(:, 0 : n+k-1) \begin{pmatrix} -Y_k^{(n)} \\ I_k \end{pmatrix} = \begin{pmatrix} 0 \\ S_k^{(n)} \\ \vdots \\ \vdots \end{pmatrix},$$

from which the formula for $x^{(n+k)}$ easily follows. Similarly, by noting that

$$s_k^{(n)} = g_0^{(n+k)}(n : n + k - 1) - \hat{H}(n : n + k - 1, 0 : n - 1)\, y_k^{(n)},$$

one obtains the formula for $y_0^{(n+k)}$. The formulas for $\hat{p}^{(n+k)}$ and $\hat{r}^{(n+k)}$ are direct consequences of the formulas for $x^{(n+k)}$ and $y_0^{(n+k)}$.

The transition from $\{x^{(n)}, y^{(n)}, \hat{p}^{(n)}, \hat{r}^{(n)}\}$ to $\{x^{(n+k)}, y^{(n+k)}, \hat{p}^{(n+k)}, \hat{r}^{(n+k)}\}$ is illustrated below. First note that

$$\begin{pmatrix} H_n & G_k & H_{0:n-1,n+k:N-1} \\ G_k^T & F_k & H_{n:n+k-1,n+k:N-1} \\ H_{n+k:2N-2,0:n-1} & H_{n+k:2N-2,n:n+k-1} & H_{n+k:2N-2,n+k:N-1} \end{pmatrix} \begin{pmatrix} -y^{(n)} & -Y_{1:k-1}^{(n)} & -y_k^{(n)} \\ 1 & 0 & 0 \\ 0 & I_{k-1} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hat{r}_{n:n+k-1}^{(n)} & \hat{S}_{n:n+k-1,1:k-1}^{(n)} & \hat{s}_k^{(n)} \\ \hat{r}_{n+k:2N-2}^{(n)} & \hat{S}_{n+k:2N-2,1:k-1}^{(n)} & \hat{S}_{n+k:2N-2,k}^{(n)} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \hat{S}_k^{(n)} & \hat{s}_k^{(n)} \\ \hat{S}_{n+k:2N-2,1:k-1}^{(n)} & \hat{S}_{n+k:2N-2,k}^{(n)} \end{pmatrix}.$$

Thus

$$\begin{pmatrix} -Y_k^{(n)} & y_k^{(n)} \\ 0 & 0 \\ I_k & 0 \end{pmatrix} \begin{pmatrix} (S_k^{(n)})^{-1} e_k & (S_k^{(n)})^{-1} s_k^{(n)} \\ 0 & 1 \end{pmatrix} = (x^{(n+k)} \quad y^{(n+k)}). \tag{4.5}$$

This can be summarized as follows:

Outline of a Single Step of Inversion Algorithm

Given the stepsize k and the vectors $\{x^{(n)}, y^{(n)}, \hat{p}^{(n)}, \hat{r}^{(n)}\}$:

1. Compute y_i, $i = 0, \ldots, k$, by the recurrence (3.10). Cost is $O(k \cdot n)$ operations.
2. Compute $\hat{s}_i$, $i = 0, \ldots, k$, by the recurrence (3.12). Cost is $O(k \cdot (N - n))$ operations.
3. Solve for u_n and v_n in (4.1). Cost is $O(k^3)$ operations.
4. Update the fundamental solutions and the corresponding residuals using (4.2)–(4.4). Cost is $O((n + k) \cdot k)$.

Remark 2. Suppose now that one also wants to solve a linear system of equations

$$Hz = b.$$

The multi-step algorithm can be extended easily to cover this case. Assume that $z^{(n)}$, the solution of the subsystem of dimension $n \times n$,

$$H_n z^{(n)} = b(0 : n - 1),$$

and the residual vector $b^{(n)}$,

$$b^{(n)} = b - H(0 : N - 1, 0 : n - 1)z^{(n)},$$

are known. Suppose that we want to solve the system of dimension $(n + k) \times (n + k)$

$$H_{n+k} z^{(n+k)} = b(0 : n + k - 1).$$

The updating formula (3.3) implies that

$$z^{(n+k)} = \begin{pmatrix} z^{(n)} \\ 0 \end{pmatrix} + \begin{pmatrix} -Y_k^{(n)} \\ I_k \end{pmatrix} w^{(n)}, \tag{4.6}$$

where $w^{(n)}$ is the solution of the system

$$S_k^{(n)} w^{(n)} = b(n : n + k - 1) - H(n : n + k - 1, 0 : n - 1)z^{(n)}.$$

Also, in view of (4.6) the residual $b^{(n+k)}$ can be updated as

$$b^{(n+k)} = b^{(n)} + \hat{S}_k^{(n)} w^{(n)}.$$

Remark 3. The multistep algorithm also provides a block triangular decomposition of the matrix H. Let $\{H_{n_i}\}_{i=1}^{K}$ be a sequence of succesive well-conditioned principal submatrices of H. Set $k_i = n_{i+1} - n_i$, $i = 1, \ldots, K - 1$. Then $\{S_{k_i}^{(n_i)}\}_{i=1}^{K}$ is the sequence of the corresponding Schur complements, and $\{Y_{k_i}^{(n_i)}\}_{i=1}^{K}$ are the matrices composed of the solutions of the corresponding generalized Yule–Walker equations. Let us define $(N - 1) \times (N - 1)$ upper triangular metrices $\{U_i\}_{i=1}^{K}$,

$$U_i = \begin{pmatrix} I_{n_i} & Y_{k_i}^{(n_i)} & 0 \\ 0 & I_{k_i} & 0 \\ 0 & 0 & I_{N-1-n_{i+1}} \end{pmatrix}.$$

Then a repeated use of (3.1) gives the following decomposition

$$H_{N-1} = U_K^T \cdot \cdot \cdot U_1^T \operatorname{diag}(S_{k_1}^{(n_1)}, \ldots, S_{k_K}^{(n_K)}) U_1 \cdot \cdot \cdot U_K,$$

which is a block triangular decomposition of H_{N-1} in a factored form.

5. Numerical Considerations

In the previous section we assumed that the matrix $S_k^{(n)}$ is well-conditioned (relative to H) and u_n and v_n are determined by a numerically stable method. These assumptions are critical for the accuracy of the multi-step method. Thus the major issue for the multi-step method and all other look-ahead methods is the determination of the stepsize k (Cabay and Meleshko, 1991; Chan and Hansen, 1990; Freund and Zha, 1992).

The accuracy of the fundamental solutions $x^{(n)}$ and $y^{(n)}$ is determined by the size of the condition number cond of H_n, $\operatorname{cond}(H_n) = \|H_n\|_1 \cdot \|H_n^{-1}\|_1$. The purpose of a multistep is to skip the computation of all $x^{(i)}$ and $y^{(i)}$ for which the condition number of H_i is significantly larger that that of $H = H_{N-1}$. If the condition number of H is known then one wants to form only those fundamental solutions $x^{(j)}$ for which the corresponding matrix H_j has the condition number comparable to that of H. Hence one wants to perform a multistep which takes as data the fundamental solutions $x^{(n)}$ and $y^{(n)}$ corresponding to a well-conditioned submatrix H_n and produces the fundamental solutions $x^{(n+k)}$ and $y^{(n+k)}$ corresponding to the next well-conditioned submatrix H_{n+k}. This idea was first proposed in Chan and Hansen (1990).

Starting with H_n, $x^{(n)}$, and $y^{(n)}$, one way of determining the size of k is by computing or estimating the condition number of H_{n+i}, $i = 0, 1, \ldots, k$, and picking H_{n+k} for which the estimate is deemed satisfactory. First, an estimate of $\|H_{n+i}\|_1$ is readily available. By letting

$$\nu_{n+i} \stackrel{\text{def}}{=} \sum_{j=0}^{2(n+j-1)} |\eta_j|,$$

we have

$$\|H_{n+i}\|_1 \le \nu_{n+i} \le 3 \cdot \|H_{n+i}\|_1.$$

In order to get an estimate for $\|H_{n+i}^{-1}\|$ some kind of knowledge of H_{n+i}^{-1} is necessary. For a Hankel matrix H_{n+i}, due to the inversion formula (2.4), the inverse is determined by the fundamental solutions $x^{(n+i)}$ and $y^{(n+i)}$. Given $x^{(n+i)}$ and $y^{(n+i)}$ the norm of the inverse of H_{n+i} can be easily esti-

mated. Let

$$t_i^{(n)} \stackrel{\text{def}}{=} (\|y_0^{(n)}\|_\infty, \ldots, \|y_i^{(n)}\|_\infty)^T.$$

Then due to (4.5),

$$\mu_x^{(n+i)} \stackrel{\text{def}}{=} \max\left(|t_i^{(n)}|^T \begin{pmatrix} |u_i^{(n)}| \\ 0 \end{pmatrix}, \|u_i^{(n)}\|_\infty\right)$$

and

$$\mu_y^{(n+i)} \stackrel{\text{def}}{=} \max\left(|t_i^{(n)}|^T \begin{pmatrix} |v_i^{(n)}| \\ 0 \end{pmatrix}, \|v_i^{(n)}\|_\infty\right)$$

are upper bounds on $\|x^{(n+i)}\|$ and $\|y^{(n+i)}\|$, respectively. Thus

$$c_{n+i} \stackrel{\text{def}}{=} \nu_{n+i} \cdot \mu_x^{(n+i)} \cdot \mu_y^{(n+i)} \tag{5.1}$$

provides an upper bound on $\text{cond}(H_{n+i})$.

For the calculation of the estimate (5.1) it is necessary to solve systems of linear equations (4.1). These can be solved either via LU decomposition with pivoting or via QR decomposition. The LU decomposition requires fewer operations but as the matrices $S_i^{(n)}$ are indefinite, updating of the LU decomposition can be unstable. Thus, the cost of solving k systems of linear equations (4.1) via LU decomposition is $O(k^4)$ operations. On the other hand, if the QR factorization of $S_i^{(n)}$ is known it can be used to compute the QR factorization of $S_{i+1}^{(n)}$ in $O(i^2)$ operations. Thus, the cost of solving k systems of linear equations via QR decomposition is $O(k^3)$ operations. Which of the two methods is less expensive depends on the exact size of k.

Of course, it is easier to estimate the condition number of the generically small matrix $S_i^{(n)}$. However, as Example 1 shows, the smallness of this number is only necessary for smallness of the condition number of H_{n+i}.

A possible difficulty with the estimate (5.1) lies in the fact that $u_i^{(n)}$ and $v_i^{(n)}$ are obtained as solutions of possibly ill-conditioned systems of linear equations. However, ill-conditioning of $S_i^{(n)}$, $i = 1, \ldots, k-1$, will most certainly be reflected in large norms of $u_i^{(n)}$ and $v_i^{(n)}$, giving rise to a large value of the estimate c_{n+i}. Thus a large condition number of H_{n+i} will be discovered. Let us give a to some extent pathological but instructive example.

EXAMPLE 2. Suppose that $H = H_N$ is given by

$$H = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 & 1 & -c \\ 0 & & & & 1 & -c & \\ \vdots & & \ddots & \ddots & \ddots & & \\ \vdots & & \ddots & \ddots & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & & & \\ 0 & 1 & -c & & & & \\ 1 & -c & & & & & \\ -c & & & & 0 & & \end{pmatrix},$$

where $c > 1$ and of moderate size. Then H_1 and H_N are well-conditioned and the matrices H_i are singular for $i = 2, \dots, N-2$. H_{N-1} is nonsingular but ill-conditioned. We put $n = 1$ and $k = N - 2$. Then the Schur complement $S := S^{(n)}_{N-2}$ and its inverse is given by

$$S = \begin{pmatrix} 0 & & & 1 \\ & & \iddots & -c \\ & \iddots & \iddots & \\ 1 & -c & & \end{pmatrix}, \qquad S^{-1} = \begin{pmatrix} c^{N-2} & \dots & c & 1 \\ \vdots & \iddots & \iddots & \\ \vdots & \iddots & & \\ c & \iddots & & \\ 1 & & & 0 \end{pmatrix}.$$

Hence the condition number of S is of order $O(c^{N-2})$. That means it may become large if N is large. It is interesting to remark that the vector $u^{(n)}_{N-2}$ is just the first unit vector; i.e., it does not indicate the ill conditioning of S. On the other hand, we have $v^{(n)}_{N-2} = -[c^{N-2}, \dots, c]^T$.

Confidence in the estimate c_{n+i} can be increased by monitoring the condition numbers of the matrices $S^{(n)}_i$, $i = 1, \dots, k$. These can be obtained by a number of available condition estimators. The additional cost of using a condition estimater will be low as long as some form of a decomposition of the matrix $S^{(n)}_i$ is available. As we already argued, such a decomposition may be needed anyhow for updating the QR factorization of $S^{(n)}_i$. If an estimate of the norm of the inverse of $S^{(n)}_i$ is available then (3.2) provides an alternative way of estimating the condition number of H_{n+i}. Such an estimation scheme which takes into account not only $\|(S^{(n)}_i)^{-1}\|$ but also $\|Y_i\|$ was originally proposed in Chan and Hansen (1990) in the context of solving indefinite Toeplitz linear systems.

In the case when an estimate c of cond(H) is know the multistep algorithm can be summarized as follows.

Algorithm

initialization

Find the first well-conditioned H_i, $i \leq N-1$. Set $n = i$.
Factorize H_n and solve for $x^{(n)}$ and $y^{(n)}$. Compute $\hat{p}_n$ and $\hat{r}_n$.
If $n = N-1$ then stop.

loop:

Factorize $S_k^{(n)}$.
Calculate $u_k^{(n)}$ and $v_k^{(n)}$.
Calculate c_{n+k}.
If $c_{n+k} > c$, $k \leftarrow k+1$.
Calculate $y_k^{(n)}$ and $\hat{s}_k^{(n)}$. Go to *loop*.
If $c_{n+k} \leq c$, calculate $x^{(n+k)}$ and $y^{(n+k)}$.
Set $n \leftarrow n+k$. If $n = N-1$ then stop.
Calculate $\hat{p}_n$ and $\hat{r}_n$. Go to *loop*.

It is easily seen that in the worst case the cost of the algorithm is $O(N^3)$ arithmetic operations but the accuracy of the algorithms is guaranteed.

In a situation where an estimate of cond(H) is not available, the algorithm has to be modified in the part where the decision for accepting the stepsize k is made. One possible heuristic is to set an upper bound on the step size k, that is, to restrict the range for searching for the next well-conditioned submatrix Chan and Hansen (1990). By setting the maximal range to $k_{\max}$ the cost of the algorithm can be controlled, however, the accuracy cannot. The accuracy can be checked by computing the residual vectors and if needed a step of iterative refinement can be applied to improve the accuracy of the fundamental solutions.

In conclusion, it appears that it is possible to control the accuracy or the cost of the multistep Hankel solver. It remains to be seen whether there exists a stable Hankel solver with $O(N^2)$ complexity.

REFERENCES

BAREISS, E. H. (1969), Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices, *Numer. Math.* **13**, 404–424.

BERLEKAMP, E. R. (1968), "*Algebraic Coding Theory*," McGraw–Hill, New York.

BOJANCZYK A. W., LEE, T. J. AND LUK, F. T. (1991), An accurate fast Hankel matrix solver, *in* "Advanced Signal Processing Algorithms, Architectures and Implementations II," Proc. of SPIE, Vol. 1566, 74–83.

CABAY S., AND MELESHKO, R., (1991), "A Weakly Stable Algorithm for Pade Approximation and the Inversion of Hankel Matrices," Tech. Rep., Department of Computer Science, University of Alberta, Canada.

CHAN, T. F., AND HANSEN, P. C. (1990), "A Stable Levinson Algorithm for General Toeplitz Systems," CAM Rep. 90-11, Computational and Applied Mathematics, University of California, Los Angeles, CA.

CYBENKO G. (1980), The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations, *SIAM J. Sci. Stat. Comput.* **1** (No. 3), 303–319.

DELSARTE, P., GENIN, Y., AND KAMP, Y. (1985), A generalization of the Levinson algorithm for Hermitian Toeplitz matrices with arbitrary rank profile, *IEEE Trans. Acoust. Speech Signal Process.* **33**, 964–971.

FREUND R. W., AND ZHA, H. (1992), "A Look-Ahead Algorithm for the Solution of General Hankel Systems," Tech. Rep., RIACS.

GUTKNECHT, M. H. (1992), "Stable Row Recurrences for the Pade Table and Generically Superfast Look-Ahead Solvers for Non-hermitian Toeplitz Systems," Tech. Rep. IPS 92-14, ETH Zurich.

HEINIG, G. (1992), Inversion of generalized Cauchy matrices and other classes of structured matrices, *in* "Proceedings of the IMA Workshop on Linear Algebra in Signal Processing," Institute for Mathematics and Its Applications, University of Minnesota, Minneapolis.

HEINIG, G., AND ROST, K. (1984), "Algebraic Methods for Toeplits-like Matrices and Operator, Birkhauser, Basel.

HOUSEHOLDER, A. (1964), "The Theory of Matrices in Numerical Analysis," Dover, New York.

KUNG, S.-Y. (1977), "Multivariable and Multidimensional Systems: Analysis and Design," Ph.D. dissertation, Department of Electrical Engineering, Stanford University, Stanford, CA.

LEVINSON, N. (1946), The Wiener RMS (root-mean-square) error criterion in filter design and prediction, *J. Math. Phys.* **25**, 261–278.

MASSEY, J. L. (1967), Shift register synthesis and BCH decoding, *IEEE Trans. Inform. Theory* **15** (1967), 122–127.

PAL, D., AND KAILATH, T. (1990), "Fast Triangular Factorization of Hankel and elated Matrices with Arbitrary Rank Profile," Tech. Rep. Information Systems Laboratory, Stanford University.

PHILLIPS, J. L. (1991), The triangular decomposition of Hankel matrices, *Math. Comp.* **25** (No. 115), 599–602.

RISSANEN, J. (1973), Algorithms for triangular decomposition of block Hankel and Toeplitz matrices with applications of factoring positive polynomials, *Math. Comp.* **27**, 147–154.

TRENCH, W. F. (1965), An algorithm for inversion of finite Hankel matrices, *J. SIAM* **13**, 1102–1107.

ZAROWSKI, C. J. (1991), Schur algorithms for Hermitian Toeplitz, and Hankel matrices with singular leading principal submatrices, *IEEE Trans. Signal Proc.* **39** (No. 11), 2464–2480.