

EFFICIENT SOLUTION OF A TOEPLITZ-PLUS-HANKEL  
COEFFICIENT MATRIX SYSTEM OF EQUATIONS

G. A. Merchant and T. W. Parks  
May 1980

E.E. Technical Report No. 8003  
(Revised April 1981)

This report prepared for submittal to IEEE  
Transactions on ASSP

EFFICIENT SOLUTION OF A TOEPLITZ-PLUS-HANKEL COEFFICIENT  
MATRIX SYSTEM OF EQUATIONS\*

G.A. Merchant  
T.W. Parks

Rice University  
Department of Electrical Engineering  
Houston, Texas 77001

1. Introduction

Frequently in signal processing one is faced with situations where a large system of linear equations, with a Toeplitz or a Hankel coefficient matrix, needs to be solved. One efficient way of solving these kinds of equations is by Levinson recursion [1]. The Levinson recursion does not require explicit storage of the Toeplitz (or Hankel) coefficient matrix and the number of multiplies required is proportional to the square of the number of unknowns.

There are situations [2,3,4,5] where the coefficient matrix can be represented as a sum of a Toeplitz and a Hankel matrix. Gragg, et.al. [2] for instance, discuss the case when an infinite cosine series expansion of a function is approximated by a rational function of cosines giving rise to a set of linear equations with a Toeplitz-plus-Hankel coefficient matrix. In the seismic area, the solution to the scattering problem in layered media under certain constraints [3] gives rise to the Gelfand-Levitan integral equation. The discrete analogue of the above integral equation is also Toeplitz-plus-Hankel system of

\* This work was supported by NSF Grant ENG78-09033.

equations. The Levinson recursion cannot be applied directly to this problem. In this paper we present a method by which a problem involving a sum of a Toeplitz matrix and a Hankel matrix can be converted to a block-Toeplitz form and can be solved by a block-Levinson recursion method.

This new method extends the savings obtained by Levinson recursion for inversion of a Toeplitz matrix to problems involving the inversion of matrices which are the sum of Toeplitz and Hankel matrices. The new algorithm reduces storage requirements and requires a number of multiplications proportional to the square of  $n$ , the number of unknowns.

## II. Notation

Let  $A$  be any matrix,  $p$  be a vector, and the matrix  $J$

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

be an  $n$  by  $n$  operator to perform a "flip" or reversal operation. Define

$A^T$  = transpose of matrix  $A(a_{ij} \rightleftharpoons a_{ji})$ .

$A^\pi$  = cross-transpose of matrix  $A$  around the main cross diagonal

$(a_{ij} \rightleftharpoons a_{n-1-j, n-1-i})$ . If  $A^\pi = A$ , then  $A$  is said to be persymmetric.

$p_+$  = vector  $p$ .

$p_-$  =  $Jp$  = vector  $p$  reordered in reverse index sequence.

$T$  = Toeplitz matrix such that  $\{T\}_{ij} = t(i-j)$ .

$H$  = Hankel matrix such that  $\{H\}_{ij} = h(i+j-n+1)$ .

Note that  $J^2 = \mathcal{I} = \text{identity}$  and  $JAJ = A^{\top\top} = A^{\top\top}$ . The operations of  $(\cdot)^{\top}$  and  $(\cdot)^{\top\top}$  commute. Further, if B is any other matrix then

$$(AB)^{\top\top} = JABJ = JAJ \cdot JBJ = A^{\top\top} B^{\top\top}.$$

T is persymmetric and H is symmetric. Also we define a  $2n$  by  $2n$  "interleaving" operator Q such that

$$\{Q\}_{ij} = \begin{cases} 1 & \text{for } i = 2r, j = r, r = 0, 1, \dots, n-1 \\ 1 & \text{for } i = 2r+1, j = n+r, r = 0, 1, \dots, n-1 \\ 0 & \text{for all other } (i,j) \text{ pairs} \end{cases}$$

It is simple to show that  $Q^{\top}Q = QQ^{\top} = \mathcal{I}$ . If we operate on a vector p with matrix Q then Q simply interleaves the sequences

$$p_r \text{ and } p_{n+r} \quad r = 0, 1, \dots, n-1.$$

In the matrix notation this is

$$Q[p_0 p_1 \dots p_n p_{n+1} \dots p_{2n-1}]^{\top} = [p_0 p_n p_1 p_{n+1} \dots p_{n-1} p_{2n-1}]^{\top}$$

### III. Conversion of a Toeplitz-Plus-Hankel Operator to a Block-Toeplitz Operator

Consider a problem

$$(T + H)p = b \quad (3.1)$$

where T and H are  $n$  by  $n$  Toeplitz and Hankel matrices, respectively, p is the vector to be determined and b is a known vector. Rewrite (3.1) as

$$Tp + Hp = b \quad (3.2)$$

Using the notation in section II, (3.2) can be written in two different forms:

$$Tp + HJ \cdot Jp = b \quad (3.3a)$$

$$JTJ \cdot Jp + JHp = Jb \quad (3.3b)$$

Since  $T$  is persymmetric ( $T^\pi = T$ ), it follows that

$$JTJ = T^{\pi\pi} = (T^\pi)^\pi = T^\pi$$

and (3.3a and b) become

$$Tp_+ + (HJ)p_- = b_+ \quad (3.4a)$$

$$T^\pi p_- + (JH)p_+ = b_- \quad (3.4b)$$

Equations (3.4a and b) can now be written in a block form

$$\begin{bmatrix} T & HJ \\ JH & T^\pi \end{bmatrix} \begin{bmatrix} p_+ \\ p_- \end{bmatrix} = \begin{bmatrix} b_+ \\ b_- \end{bmatrix} \quad (3.5)$$

Since  $H$  is a Hankel matrix  $H^\pi = H$  and  $(HJ)^\pi = J^\pi H^\pi = JH$ . Denoting  $T_H = HJ$ ,

(3.5) becomes

$$\begin{bmatrix} T & T_H \\ T_H^\pi & T^\pi \end{bmatrix} \begin{bmatrix} p_+ \\ p_- \end{bmatrix} = \begin{bmatrix} b_+ \\ b_- \end{bmatrix} \quad (3.6)$$

We now make a crucial observation, that  $T_H$  is a Toeplitz matrix. Thus, each block matrix in eq. (3.6) is a Toeplitz matrix. The coefficient matrix in eq. (3.6) is a  $2n$  by  $2n$  matrix. The known and unknown vectors are of length  $2n$ . Using the interleaving operator  $Q$  on eq. (3.6)

$$Q \cdot \begin{bmatrix} T & T_H \\ T_H^\pi & T^\pi \end{bmatrix} \cdot Q^\pi \cdot Q \cdot \begin{bmatrix} p_+ \\ p_- \end{bmatrix} = Q \cdot \begin{bmatrix} b_+ \\ b_- \end{bmatrix} \quad (3.7)$$

If we now define a set of 2 by 2 matrices

$$R(i,j) = \begin{bmatrix} t(i,j) & h(i,j) \\ h(j,i) & t(j,i) \end{bmatrix} \quad (3.8)$$

where  $t(i,j) = \{T\}_{ij}$  and  $h(i,j) = \{T_H\}_{ij}$  it can be seen that eq. (3.7)

gives a new coefficient matrix, in which the elements are 2 by 2

matrices similar to those defined in eq. (3.8). Since  $T$  and  $T_H$  are Toeplitz matrices

$$t(i,j) = t_{i-j} \text{ and } h(i,j) = h_{i-j}.$$

Hence,

$$R(i,j) = \begin{bmatrix} t(i-j) & h(i-j) \\ h(j-i) & t(j-i) \end{bmatrix} = R_{i-j}.$$

Using the above notation for  $R(i,j)$ , (3.7) can be written as

$$\begin{bmatrix} R_0 & R_{-1} & R_{-2} & \cdot & \cdot & \cdot & R_{-n+1} \\ R_1 & R_0 & R_{-1} & \cdot & \cdot & \cdot & R_{-n+2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{n-1} & \cdot & \cdot & \cdot & \cdot & \cdot & R_0 \end{bmatrix} \begin{bmatrix} \bar{p}_0 \\ \bar{p}_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \bar{p}_{n-1} \end{bmatrix} = \begin{bmatrix} \bar{b}_0 \\ \bar{b}_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \bar{b}_{n-1} \end{bmatrix} \quad (3.9)$$

where

$$\bar{p}_i = \begin{bmatrix} p_i \\ p_{n-1-i} \end{bmatrix} \text{ and } \bar{b}_i = \begin{bmatrix} b_i \\ b_{n-1-i} \end{bmatrix}$$

Equation (3.9) has a block-Toeplitz coefficient matrix and can be solved by a block-Levinson algorithm as shown by Akaike [6].

#### IV. Block-Levinson Algorithm

In this section a regular block-Levinson algorithm resolve (3.9) is presented. All the lower case letters represent length-2 vectors, and all the capital letters represent  $2 \times 2$  matrices.

$$1) X_0 = Y_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, P_0 = P_1 = R_0^{-1} \bar{b}_0, V_x = V_y = R_0$$

2) For  $i = 1, 2, \dots, n - 1$

$$1) E_x = \sum_{j=0}^{i-1} R_{i-j} X_j, E_y = \sum_{j=1}^i R_{+j} Y_{i-j}$$

$$2) \bar{e}_p = \sum_{j=0}^{i-1} R_{i-j} \bar{p}_j$$

$$3) B_x = V_y^{-1} E_x, B_y = V_x^{-1} E_y$$

$$4) \begin{bmatrix} \mathcal{L} \\ X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_i \end{bmatrix} \leftarrow \begin{bmatrix} \mathcal{L} \\ X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_{i-1} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ Y_{i-1} \\ \cdot \\ \cdot \\ \cdot \\ Y_1 \\ \mathcal{L} \end{bmatrix} \cdot B_x, \quad \begin{bmatrix} Y_i \\ Y_{i-1} \\ \cdot \\ \cdot \\ \cdot \\ Y_1 \\ \mathcal{L} \end{bmatrix} \leftarrow \begin{bmatrix} 0 \\ Y_{i-1} \\ \cdot \\ \cdot \\ \cdot \\ Y_1 \\ \mathcal{L} \end{bmatrix} - \begin{bmatrix} \mathcal{L} \\ X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_{i-1} \\ 0 \end{bmatrix} \cdot B_y$$

$$V_x \leftarrow V_x - E_y B_x, V_y \leftarrow V_y - E_x B_y,$$

$$5) \bar{g} = V_y^{-1} (\bar{b}_i - \bar{e}_p)$$

$$6) P_{i+1} = \begin{bmatrix} \bar{p}_0 \\ \bar{p}_1 \\ \cdot \\ \cdot \\ \bar{p}_{i-1} \\ \bar{p}_i \end{bmatrix} \leftarrow \begin{bmatrix} \bar{p}_0 \\ \bar{p}_1 \\ \cdot \\ \cdot \\ \bar{p}_{i-1} \\ 0 \end{bmatrix} + \begin{bmatrix} Y_i \\ Y_{i-1} \\ \cdot \\ \cdot \\ Y_1 \\ \mathcal{L} \end{bmatrix} \cdot \bar{g}$$

3) rearrange  $P_n = \{p_0 p_1 p_2 \dots p_{2n-1}\}^T$

$$P_r \leftarrow P_{2r}, r = 0, 1, \dots, (n - 1).$$

At the completion of step 3 the first half of the rearranged vector  $\bar{p}_n$  gives the solution to eq. (3.1). In the algorithm it is assumed that there is a unique solution to eq. (3.1), i.e.  $(T + H)$  is nonsingular. Also, in steps (2.3) and (2.5) it is assumed that the necessary inverses of  $V_x$  and  $V_y$  exist. This is true if the corresponding block-submatrices are nonsingular. This can be shown as follows:

In step 2, at the  $k^{\text{th}}$  substep, let the auxiliary vectors be denoted as

$$[\varrho, (Y_1^k)^\tau, (Y_2^k)^\tau, \dots, (Y_k^k)^\tau]^\tau$$

and

$$[\varrho, (X_1^k)^\tau, (X_2^k)^\tau, \dots, (X_k^k)^\tau]^\tau$$

and matrices  $V_x$  and  $V_y$  as  $V_x^k$  and  $V_y^k$ . Then, at the end of the  $k^{\text{th}}$  substep

the following relationships hold.

$$\begin{bmatrix} \varrho & & & & & & & & \\ X_1^1 & \varrho & & & & & & & \\ X_2^2 & X_1^2 & \varrho & & & & & & \\ X_3^3 & X_2^3 & X_1^3 & \varrho & & & & & \\ \cdot & \cdot & \cdot & \cdot & \varrho & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \varrho & & & \\ X_k^k & X_{k-1}^k & X_{k-2}^k & \cdot & \cdot & \cdot & \varrho & & \end{bmatrix} \begin{bmatrix} R_0 & R_{-1} & \cdot & R_{-k} \\ R_1 & R_0 & \cdot & R_{-k+1} \\ R_2 & R_1 & \cdot & R_{-k+2} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R_k & R_{k-1} & \cdot & R_0 \end{bmatrix} = \begin{bmatrix} V_x^0 & S_1^0 & S_2^0 & \cdot & S_k^0 \\ \cdot & V_x^1 & S_2^1 & \cdot & S_k^1 \\ \cdot & \cdot & V_x^2 & \cdot & S_k^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & V_k^k \end{bmatrix} \quad (4.1)$$

where  $S_j^i = \sum_{r=0}^i X_r^i R_{(i-j)-r}$ . For our purpose, the S matrices do not play any

significant role. Denote the matrix

$$\begin{bmatrix} R_0 & R_{-1} & \cdot & R_{-k} \\ R_1 & R_0 & \cdot & R_{-k+1} \\ R_2 & R_1 & \cdot & R_{-k+2} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R_k & R_{k-1} & \cdot & R_0 \end{bmatrix} = R_k.$$



Then, from eq. (4.1)

$$\det(\mathcal{R}_k) = \prod_{i=0}^k \det(V_x^i) \quad (4.2a)$$

and similarly it can be shown that

$$\det(\mathcal{R}_k) = \prod_{i=0}^k \det(V_y^i) \quad (4.2b)$$

However, it can be shown that [5]

$$\det(\mathcal{R}_k) = \det(T_k + H_k) \det(T_k - H_k) \quad (4.3)$$

where  $T_k$  is the leading block-submatrix along the main diagonal and  $H_k$  is the leading block-submatrix along the main cross diagonal. Clearly, if

$\det(V_x^k) = 0$ , then  $\det(T_k + H_k) = 0$  or  $\det(T_k - H_k) = 0$ , or both are zero.

Assuming that  $\det(T_k + H_k) \neq 0$  for all  $k$  the invertibility of  $\mathcal{R}_k$  depends on

whether  $\det(T_k - H_k)$  is zero or not. We shall now prescribe an approach

which in general should make the invertibility of  $\mathcal{R}_k$  independent of  $\det(T_k - H_k)$ .

Let  $D_n$  be an  $n$  by  $n$  matrix such that it is both Toeplitz and Hankel.

Then eq. (3.1) can be written as

$$(T' + H')p = b \quad (4.4)$$

where  $T' = T + D_n$  and  $H' = H - D_n$  are Toeplitz and Hankel matrices, respectively.

Clearly, we are still solving the same problem. One simple way of

generating a matrix  $D_n$  which is both Toeplitz and Hankel is to generate a

Toeplitz matrix with the vector

$$(d_{-n+1}, d_{-n+2}, \dots, d_{-1}, d_0, d_1, \dots, d_{n-1})^T$$

where

$$d_{2r} = d_{-2r} = d_0, \quad d_{2r-1} = d_{-2r+1} = d_1, \quad r = 1, 2, \dots, \text{integer } \left[ \frac{n}{2} \right]$$

and where  $\{D_n\}_{ij} = d_{i-j}$ . Here  $d_0$  and  $d_1$  can be any two numbers. Let at

the  $k^{\text{th}}$  substep in the step two of the algorithm,  $U_k$  be the leading  $(k+1)$  by  $(k+1)$  submatrix along the main diagonal of  $D_n$ , and  $W_k$  be the leading  $(k+1)$  by  $(k+1)$  submatrix along the main cross diagonal of  $D_n$  as shown below

$$D_n = \begin{bmatrix} d_0 & d_1 & d_0 & d_1 & \cdot & \cdot & \cdot \\ d_1 & d_0 & d_1 & d_0 & \cdot & \cdot & \cdot \\ d_0 & d_1 & d_0 & d_1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} = \begin{bmatrix} U_k & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} = \begin{bmatrix} \cdot & V_k \\ \cdot & \cdot \end{bmatrix}$$

Both  $U_k$  and  $W_k$  have the same property as  $D_n$ , viz. each of them is both Toeplitz and Hankel. If  $|d_0| \neq |d_1|$  and if the number of unknowns  $n$  is odd then for  $k = 0, 2, 4, \dots$

$$U_k = W_k, U_{k+1} \neq W_{k+1}.$$

If  $|d_0| \neq |d_1|$  and if  $n$  is even then for  $k = 0, 2, 4, \dots$

$$U_k \neq W_k, U_{k+1} = W_{k+1}.$$

If  $d_0 = d_1$  then for  $k = 0, 1, 2, \dots$

$$U_k = W_k.$$

If  $d_0 = -d_1$  then for  $k = 0, 1, 2, \dots$

$$U_k = \pm W_k.$$

the sign being dependent on the values of  $n$  and  $k$ . The equation (4.3) now becomes

$$\det(\mathcal{R}'_k) = \det(T'_k + H'_k) \det(T'_k - H'_k)$$

where  $T'_k = T_k + U_k$  and  $H'_k = H_k + W_k$ . Even if  $T_k - H_k$  is singular, addition of an arbitrary matrix  $U_k + W_k$  will in general make it nonsingular. For the

same reason,  $T_k + H_k$  which has been assumed to be nonsingular, will remain so.

This, in turn, implies that in general  $\det(R'_k) \neq 0$  which ensures that in

$$(4.2) \quad \prod_{i=0}^k \det(V'_x{}^i) = \prod_{i=0}^k \det(V'_y{}^i) \neq 0. \quad \text{Hence, } V'_x{}^k \text{ and } V'_y{}^k \text{ are invertible.}$$

If during execution of the algorithm, we encounter  $V'_x{}^k$  and  $V'_y{}^k$  with determinant zero, we need to restart the algorithm from the beginning with modified  $T$  and  $H$ . A convenient way of avoiding this is to start with modified  $T$  and  $H$  and solve the problem, regardless of whether the unmodified problem is known to be solvable or not. One simple choice of matrix  $D_n$  is to use  $d_0 = t_0$  and  $d_1 = 0$ . This modified procedure has been implemented with excellent results.

#### V. Use of Symmetries to Reduce Computation

Write (3.9) as

$$R_n P = B \quad (5.1)$$

where  $R_n$  is the block-Toeplitz coefficient matrix. We show that  $R_n^{\tau\pi} = R_n$ .

Since  $R_n^{\tau\pi} = J_{2n} R_n J_{2n}$ , the  $(i,j)$ <sup>th</sup> block element of  $R_n^{\tau\pi}$  is

$$\{R_n^{\tau\pi}\}_{ij} = J_2 R_{(n-1-i, n-1-j)} J_2 = J_2 R_{j-i} J_2 = R_{j-i}^{\tau\pi} \quad (5.2)$$

But,

$$R_{j-i} = \begin{bmatrix} t_{j-i} & h_{j-i} \\ h_{i-j} & t_{i-j} \end{bmatrix}$$

hence

$$R_{j-i}^{\tau\pi} = \begin{bmatrix} t_{i-j} & h_{i-j} \\ h_{j-i} & t_{j-i} \end{bmatrix} = R_{i-j}.$$

Thus,  $\{R_n^{\tau\pi}\}_{ij} = \{R_n\}_{ij}$ . Also, note that  $J_{2n} P = P$  and  $J_{2n} B = B$ .

Using the above symmetries, it can be shown in the algorithm of section

IV that the following relations hold:

$$a) E_y = E_x^{\top\pi}$$

$$b) B_y = B_x^{\top\pi}$$

$$c) Y_r = X_r^{\top\pi}, r = 0, 1, 2, \dots, i$$

$$d) V_y = V_x^{\top\pi}$$

These relations lead to a simplified algorithm, requiring less computation and storage. The new algorithm is:

$$1) X_0 = Y_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, P_0 = P_1 = R_0^{-1} \bar{b}_0, V_x = R_0$$

2) for  $i = 1, 2, \dots, n - 1$

$$1) E_x = \sum_{j=0}^{i-1} R_{i-j} X_j$$

$$2) \bar{e}_p = \sum_{j=0}^{i-1} R_{i-j} \bar{p}_j$$

$$3) B_x = (V_x^{\top\pi})^{-1} E_x$$

$$4) \begin{bmatrix} \mathcal{L} \\ X_1 \\ X_2 \\ \cdot \\ \cdot \\ \cdot \\ X_i \end{bmatrix} \leftarrow \begin{bmatrix} \mathcal{L} \\ X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_{i-1} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ X_{i-1}^{\top\pi} \\ \cdot \\ \cdot \\ \cdot \\ X_1^{\top\pi} \\ \mathcal{L} \end{bmatrix} \cdot B_x$$

$$V_x \leftarrow V_x - E_x^{\top\pi} B_x,$$

$$5) \bar{g} = (V_x^{\top\pi})^{-1} (\bar{b}_i - \bar{e}_p)$$

$$6) \bar{p}_{i+1} = \begin{bmatrix} \bar{p}_0 \\ \cdot \\ \cdot \\ \bar{p}_{i-1} \\ \bar{p}_i \end{bmatrix} - \begin{bmatrix} \bar{p}_0 \\ \cdot \\ \cdot \\ \bar{p}_{i-1} \\ 0 \end{bmatrix} + \begin{bmatrix} x_i^{\tau\pi} \\ x_{i-1}^{\tau\pi} \\ \cdot \\ \cdot \\ \cdot \\ x_1^{\tau\pi} \\ \rho \end{bmatrix} \cdot \bar{g}$$

$$3) \text{ rearrange } \bar{p}_n = \{p_0 p_1 p_2 \dots p_{2n-1}\}^{\tau}$$

$$p_r \leftarrow p_{2r}, \quad r = 0, 1, \dots, (n-1).$$

The remarks following the algorithm in section IV hold here too.

## VI. Conclusions

The coefficient matrix in (3.5) belongs to a more general class of matrices called the centrosymmetric matrices. Given two matrices A and B we can form a centrosymmetric matrix as

$$\begin{bmatrix} A & BJ \\ JB & JAJ \end{bmatrix}.$$

Good [5] has shown that finding the inverse of the above matrix is equivalent to finding inverses of two smaller matrices, viz.  $(A + B)$  and  $(A - B)$ . In our paper we have taken the opposite view and created a larger centrosymmetric matrix which, because of the special properties of T and H, can be inverted more efficiently than inverting  $(T + H)$  directly.

The method described in this paper has been used to solve a system of equations with 255 unknowns on a PDP 11/55 in double precision. The execution time required was about one minute. For the purposes of com-

parison a simplified version of the same problem (where matrix  $T$  turns out to be identity) was solved by the Conjugate Gradient (CG) method [7] for symmetric matrices. The CG method required six major iterations to solve the problem with comparable accuracy and the execution time was of the order of four hours. The CG method was chosen for comparison since, like the method described in this paper, it does not require explicit storage of the coefficient matrix. Moreover the total array size required is comparable in each case. The results are not surprising considering the CG method has the computational complexity of  $O(n^3)$ .

## References

- [1] Levinson, N., "The Wiener RMS (root mean square) Error Criterion in Filter Design and Prediction," J. Math. and Phys., Vol. 25, pp. 261-278 (1947).
- [2] Gragg, W.B. and Johnson, G.D., "The Laurent-Pade Table," Proceedings of IFIP Congress 1974, pp. 632-637, North-Holland Publishing Co. (1974).
- [3] Aki, K. and Richards, P.G., Quantitative Seismology Theory and Methods, W.H. Freeman and Co., San Francisco (1980).
- [4] Dugre, J., Beex, A.A.L., and Scharf, L.L., "Generating Covariance Sequences and Calculation of Quantization and Rounding Error Variances in Digital Filters," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-28(1), pp. 102-104 (February 1980).
- [5] Good, I.J., "The Inverse of a Centrosymmetric Matrix," Technometrics, Vol. 12(4), pp. 925-928 (November 1970).
- [6] Akaike, H., "Block Toeplitz Matrix Inversion," SIAM J. Appl. Math., Vol. 24(2), pp. 234-241 (March 1973).
- [7] Hestenes, M.R. and Stiefel, E., "Methods of Conjugate Gradient for Solving Linear Systems," Journal of Res. of Nat. Bureau of Standards, Vol. 49(6), pp. 407-436 (December 1952).