The Solution of a Special
Set of Hermitian Toeplitz Linear Equations

by

David C. Farden

ONR Technical Report #12
August 1975

Prepared for the Office of Naval Research
under Contract N00014-75-C-0518

L. L. Scharf and M. M. Siddiqui, Principal Investigators

Reproduction in whole or in part is permitted
for any purpose of the United States Government

Approved for public release; distribution unlimited.
The Solution of a Special Set of Hermitian Toeplitz Linear Equations

by

David C. Farden

Electrical Engineering Department
Colorado State University
Fort Collins, Colorado 80523

January 1975

Submitted to

ACM Transactions on Mathematical Software

This work was supported by the Office of Naval Research under Grants N00014-67-A-0299-0019, and N00014-75-C-0518, and the Naval Undersea Center under Contracts N66001-74-C-0035 and N66001-75-C-0224.
ABSTRACT

The solution of a set of \( \mathbf{m} \) linear equations \( \mathbf{L} \mathbf{s} = \mathbf{d} \), where \( \mathbf{L} \) is an \( \mathbf{m} \)-th order Hermitian Toeplitz matrix and the elements of \( \mathbf{d} \) possess a Hermitian symmetry, is considered. A specialized algorithm is developed for this case which solves for \( \mathbf{s} \) in approximately \( 1.5\mathbf{m}^2 \) "operations," whereas the Hermitian case of an algorithm developed by Zohar solves for \( \mathbf{s} \) in approximately \( 2\mathbf{m}^2 \) "operations." An "operation" is used here to denote one addition and one multiplication. A further reduction in computational requirements is shown in case \( \mathbf{L} \) and \( \mathbf{d} \) are real. As with Zohar's algorithm, the specialized algorithm requires that all principal minors of \( \mathbf{L} \) be nonzero.

KEY WORDS AND PHRASES: Linear algebra, linear equations, Toeplitz matrix, computer programming

CR CATEGORIES: 5.14, 5.25
1. Introduction

Consider the set of linear equations

\[ L_m s_m = d_m. \] (1)

Zohar [1] makes use of the Trench algorithm [2], [3] to develop an efficient algorithm for solving (1) when \( s_m, d_m \) are \( mx1 \) matrices and \( L_m \) is a non-Hermitian \( \text{mth-} \)order Toeplitz matrix. In this paper, an efficient algorithm is developed for solving (1) when \( L_m \) is a Hermitian Toeplitz matrix and \( d_m \) satisfies

\[ d_m^* = d_m. \] (2)

where the symbol \(^*\) is used to denote the reversed ordering of the elements of \( d_m \), i.e., \( (d_m)_{1,l}^* = (d_m)_{m+1-l,l} \) and \(^*\) denotes complex conjugate. Such a specialized case can arise, for example, in the design of digital filters, as discussed in [4]. The following example serves to illustrate how such a system of equations can arise.

EXAMPLE. Let \( \alpha(t), \beta(t), \gamma(t) \) be jointly wide-sense stationary complex-valued stochastic processes with \( \alpha(t) = \beta(t) + \gamma(t) \), where \( \mathbb{E}\{\beta(t)\gamma^*(s)\} = 0 \) for all real \( t \) and \( s \) and \( \mathbb{E}\{\cdot\} \) denotes statistical expectation. On the basis of the observation vector \( a_m(k), a_m(k) = [a(k), a(k-1), \ldots, a(k-m+1)] \), where the symbol \(^\top\) denotes matrix transpose, it is desired to compute a linear minimum mean-square error (MMSE) estimate of \( \beta(k-p) \), i.e., it is desired to minimize the quantity \( \mathbb{E}\{|a_m(k) - \beta(k-p)|^2\} \) with respect to \( a_m \).

It is easily shown that the desired solution, \( s_m \), satisfies (1), with

\[ L_m = \mathbb{E}\{a_m(k)^*a_m(k)\} \quad \text{and} \quad d_m = \mathbb{E}\{(\beta(k-p)a_m^*)\}. \]

Since \( \alpha(t) \) is wide-sense stationary, \( L_m \) is a Hermitian Toeplitz matrix. With \( p = (m+1)/2 \), it is easily seen that (2) is satisfied since \( \beta(t) \) and \( \gamma(t) \) are jointly wide-sense
stationary.

A useful consequence of the assumptions that \( \hat{d}_m = d_m \) and \( \hat{L}_m = L_m \) is that \( \hat{s}_m = s_m \). Define \( E_m \) to be the \( m \times m \) exchange matrix of Zohar [3], i.e.,

\[
E_m a = a
\]

for any \( m \times 1 \) matrix \( a \). Note that \( E_m E_m = I_m \), where \( I_m \) is the \( m \times m \) identity matrix. Since \( L_m \) is persymmetric [3], \( E_m L_m E_m = \bar{L}_m \). Since \( d_m = \bar{d}_m \) from (1) we have \( E_m L_m (E_m E_m) s = \bar{L}_m \hat{s}_m \), so that \( \hat{s}_m = \bar{L}_m \hat{s}_m \), i.e.,

\[
\hat{s}_m = s_m
\]

The specialized algorithm developed in Section 3 of this paper solves (1) with \( d_m \) satisfying (2) in approximately \( 1.5m^2 \) complex "operations," whereas the Hermitian case of Zohar's algorithm [1] uses approximately \( 2m^2 \) complex "operations." An "operation" is used here to denote one addition and one multiplication. In case \( L_m d_m \) (and hence \( s_m \)) are real, the results of Section 3 can be used to solve (1) in approximately \( 1.25m^2 \) real multiplications and \( 1.5m^2 \) real additions.

Both Zohar's algorithm [1] and the specialized algorithm developed in Section 3 make use of Phase 1 of the Trench algorithm [1]-[3]. Rather than review the results necessary for the development of Section 3, it is assumed that the reader is familiar with the work of Zohar [3].

2. Preliminaries

Since the techniques used in this paper are inherently related to those used by Zohar [1], an attempt is made to follow the same notational conventions. Greek letters are used for scalars, capital letters for square matrices, and lower-case letters for column matrices. Subscripts used on matrices are used to denote the number of elements in one column of the matrix.

Since Phase 1 of the Trench algorithm requires that all principal minors of \( L_m \) be nonzero, it is assumed that (1) has been normalized so that \( L_m \) has ones along its main diagonal.
3. The Specialized Algorithm

Consider the system of equations $L_m s = d_m$ where $L_m$ is an $m$th order normalized Hermitian Toeplitz matrix and $d_m = \hat{d}_m$, so that $d_m$ may be written as $d_m = \left[ \xi_1 \frac{\xi_2 + \xi_2^*}{2} \xi_1 \xi_2 \cdots \xi_1 \xi_2^* \cdots \xi_1 \frac{\xi_m + \xi_m^*}{2} \right]$ for $m$ odd and $d_m = \left[ \frac{\xi_1}{2} \frac{\xi_1}{2} 1 1 \frac{\xi_m}{2} \frac{\xi_m}{2} \right]$ for $m$ even. For $m$ even or odd we may write $\hat{d}_m = \left[ \xi_{i+1}^* \xi_i \frac{\xi^*}{2} \cdots \frac{\xi^*}{2} \right]$, for $i=1,2,\cdots,m-2$ where $[x]$ denotes the largest integer less than or equal to $x$. The Hermitian Toeplitz nature of $L_m$ enables us to write

$$L_{i+2} = \begin{bmatrix} 1 & r_{i+1} & \hat{r}_{i+1} \\ r_{i+1}^* & L_{i+1} & \hat{r}_{i+1} \\ \hat{r}_{i+1} & L_{i+1} & 1 \end{bmatrix}, \quad (3)$$

where $r_{i+1} = [\rho_1 \rho_2 \cdots \rho_{i+1}] (0 \leq i \leq m-2)$. Clearly, (3) may be rewritten as

$$L_{i+2} = \begin{bmatrix} 1 & r_i & \hat{r}_{i+1} \\ r_i^* & L_i & \hat{r}_{i+1} \\ \hat{r}_{i+1} & L_i & 1 \end{bmatrix}.$$

Defining $L_{i+2} s_{i+2} = \hat{d}_{i+2}$ (1 \leq i \leq m-2), we have $L_{i+2} \begin{bmatrix} s_{i+2} \\ s_i \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_0 \\ \theta_1 \end{bmatrix}$, where $\theta_0 = \xi_{i+3} \frac{\xi_i}{2} - r_is_i$ and $\theta_1$ is an $i \times 1$ column matrix of zeros.
Defining \( B_{i+2} = s_{i+2}^{-1} \), we obtain

\[
s_{i+2} = \begin{bmatrix} 0 \\ s_i \\ 0 \end{bmatrix} + B_{i+2} \begin{bmatrix} \theta_i \\ 0 \\ \theta_i \end{bmatrix}
\]

(4)

Since the inverse of a Hermitian persymmetric matrix is a Hermitian persymmetric matrix [3], \( B_{i+2} \) may be expressed in the form

\[
B_{i+2} = \lambda_{i+1}^{-1} \begin{bmatrix} 1 & e_{i+1}^* \\ * & e_{i+1} M_{i+1} \end{bmatrix} = \lambda_{i+1}^{-1} \begin{bmatrix} P_{i+1} & e_{i+1}^* \\ * & e_{i+1} \end{bmatrix}.
\]

Letting \( f_i = [I, 0] e_{i+1} \), we may write

\[
B_{i+2} = \lambda_{i+1}^{-1} \begin{bmatrix} 1 & f_i \\ * & e_{i+1} \end{bmatrix}.
\]

(5)

Substituting (5) into (4) we obtain the result

\[
s_{i+2} = \begin{bmatrix} 0 \\ s_i \\ 0 \end{bmatrix} + \lambda_{i+1}^{-1} \theta_i \begin{bmatrix} 1 \\ * \\ e_{i+1} \end{bmatrix} + \theta_i^{-1} \begin{bmatrix} e_{i+1}^* \\ * \\ 1 \end{bmatrix}.
\]

(6)

In order to make use of this result, we apply the recursive relationships for Phase 1 of the Trench algorithm [1]:

Initial values: \( e_1 = -\rho_1, \lambda_1 = 1 - |\rho_1|^2 \)
Recursive relationships: \( \eta'_i = \rho_{i+1} e_i r_i, \)

\[
e_{i+1}' = \begin{bmatrix} e_i + \eta_i \lambda_{i+1}' \xi_i \n + \eta_i \lambda_{i+1}' \xi_i \end{bmatrix}, \quad \lambda_{i+1}' = \lambda_i - |\eta_i|^2 \lambda_i^{-1}.
\]

Finally, Phase 1 of the Trench algorithm and (6) may be combined by noting that

\[ s_1 = \xi_1 \] (7)

and

\[ s_2 = (1 - |\rho_1|^2)^{-1} \begin{bmatrix} \xi_1 - \rho_1 \xi_1 \n + \rho_1 \xi_1 \end{bmatrix}. \] (8)

An immediate consequence of (6), (7), and (8) is that \( s_{1+2} = s_{i+2} \)
since \( \lambda_{i+1}' \) is real-valued. Consequently, there are two sources of increased computational speed in the specialized algorithm: (1) \( s_{1+2} \) need only be computed for \( i = 1, 3, 5, \ldots, m-2 \) when \( m \) is odd and for \( i = 2, 4, 6, \ldots, m-2 \) when \( m \) is even, and (ii) approximately half \( (\frac{i+3}{2}) \) of the elements of \( s_{1+2} \) need to be computed using (6), the remaining elements being obtained from the relationship \( s_{1+2} = s_{i+2} \). The following is a summary of the algorithm.

**PROBLEM FORMULATION:** \( L_s = d_m, L_m = \begin{bmatrix} r_1 \n m \n m \end{bmatrix} \),

\( r_{1=m-1} \) when \( m \) is odd,

\[ d_1 \] \begin{bmatrix} \xi_1 \n \xi_1 \end{bmatrix} (1 \leq i \leq m-1),

\[ d_{i+2} = \begin{bmatrix} \xi_{i+3} \n \xi_{i+3} \end{bmatrix}, \quad s_m = ? \]
Initial values: \( e_1 = -\rho_1, \quad \lambda_1 = 1 - |\rho_1|^2 \),

\[
\begin{bmatrix}
\xi_1 - \rho_1 \\
\xi_1^* - \rho_1^*
\end{bmatrix}
\]

Recursive relations: Compute \( \eta_i, \quad e_{i+1}, \quad \text{and} \quad \lambda_{i+1} \) for \( i = 1, 2, \ldots, m-2 \).

Compute \( \theta_i \) and \( s_{i+2} \) for \( i = 1, 3, 5, \ldots, m-2 \) for \( m \) odd and

\( i = 2, 4, 6, \ldots, m-2 \) for \( m \) even.

\[
\eta_i = -\rho_{i+1} - e_i r_i
\]

\[
e_{i+1} = \begin{bmatrix} e_i + \eta_i \lambda_i \hat{e}_i \\ \eta_i \lambda_i \end{bmatrix}
\]

\[
\lambda_{i+1} = \lambda_i - |\eta_i \lambda_i|^{1/2}
\]

\[
\theta_i = \frac{\xi_{i+1}}{i+3} - \frac{r_i s_i}{2 i
\]

\[
s_{i+2} = \begin{bmatrix} 0 \\ s_i \end{bmatrix} + \lambda_{i+1}^{-1} \begin{bmatrix} 1 \\ e_{i+1} \end{bmatrix} + \theta_{i+1} \begin{bmatrix} 1 \\ e_{i+1} \end{bmatrix}
\]

Making use of the fact that only \( \frac{i+3}{2} \) elements of \( s_{i+2} \) need be computed, the above algorithm requires approximately \( 1.5m^2 \) additions and \( 1.5m^2 \) multiplications for the solution of \( s_m \). This compares with \( 2m^2 \) for the Hermitian case of Zohar's algorithm [1].

In case \( L_m, \quad d_m \) (and hence \( s_m \)) are real, an even further reduction in computational requirements results. For this case (6) may be rewritten as

\[
s_{i+2} = \begin{bmatrix} 0 \\ s_i \end{bmatrix} + \lambda_{i+1}^{-1} \begin{bmatrix} 1 \\ e_{i+1} \end{bmatrix} + \hat{e}_{i+1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad (9)
\]
and the computation of $s_{1+1}$ in the expression for $\theta_1$ may be computed as

$$r_1 s_{1} = \sum_{l=1}^{1/2} (s_{1})^l (\rho_{l+1}^{1/2})^{1-1/2}$$

(10)

for $i$ even and

$$r_1 s_{1} = \sum_{l=1}^{i-1/2} (s_{1})^l (\rho_{l+1}^{1/2})^{1-1/2} + \frac{s_{1}^{i+1}}{2}$$

(11)

for $i$ odd. Making use of these expressions, the specialized algorithm requires approximately $1.5m^2$ additions and $1.25m^2$ multiplications. A slightly different form of (9) can be easily obtained as

$$s_{1+2} = \begin{bmatrix} 0 \\ s_1 \\ 0 \end{bmatrix} + \frac{\theta_{1}}{\lambda_{1}^{-1}} \begin{bmatrix} 1 \\ e_{1}^{r_{1}} + e_{1} \\ 1 \end{bmatrix}$$

(12)

This final expression (12) is slightly more efficient than (9). A FORTRAN routine for the specialized algorithm making use of (10)-(12) is presented in [5].

EXAMPLE. Let $\rho_{i} = (i+1)^{-1}$ for $i=1,2,\ldots, m-1$ and $\xi_{i} = i^{-1}$, for $i=1,2,\ldots, \lceil \frac{m+1}{2} \rceil$. A FORTRAN routine, called TPSLV, based on the symmetric case of [1] was written for a timing comparison with the FORTRAN routine, called SYMM, presented in [5]. The time needed (in seconds) for each routine to compute $s_{m}$ for this example with $m \in \{10,50,100,500\}$ is indicated in the following table.

<table>
<thead>
<tr>
<th>$M$</th>
<th>TPSLV</th>
<th>SYMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>.005</td>
<td>.005</td>
</tr>
<tr>
<td>50</td>
<td>.089</td>
<td>.057</td>
</tr>
<tr>
<td>100</td>
<td>.343</td>
<td>.217</td>
</tr>
<tr>
<td>500</td>
<td>8.266</td>
<td>5.233</td>
</tr>
</tbody>
</table>

The above results, obtained on a CDC 6400 computer, agree with the computational considerations presented above.
4. Concluding Remarks

An algorithm has been developed for the solution of a specialized set of Toeplitz linear equations that arise in linear filtering applications. The savings in computational requirements of the new algorithm over the results of Zohar [1] are approximately 25% for the Hermitian case and 37.5% for the real case. Finally, it is noted that the techniques used in developing the specialized algorithm can indeed be applied to the general case treated by Zohar [1]; however, such a development results in an algorithm having no computational advantage over the generalized algorithm of [1].
REFERENCES


The Solution of a Special Set of Hermitian Toeplitz Linear Equations

David C. Farden

Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80523

Office of Naval Research, Code 436
Statistics and Probability Branch
Arlington, VA 22217

August 1975

Approved for public release; distribution unlimited.

Linear algebra; Linear equations; Toeplitz matrix; Computer programming.

The solution of a set of linear equations \( L \mathbf{x} = \mathbf{b} \), where \( L \) is an \( m \times m \) order Hermitian Toeplitz matrix and the elements of \( \mathbf{b} \) possess a Hermitian symmetry, is considered. A specialized algorithm is developed for this case which solves for \( \mathbf{x} \) in approximately \( \frac{3}{2} m^2 \) operations, whereas the Hermitian case of an algorithm developed by Zohar solves for \( \mathbf{x} \) in approximately \( m^2 \) operations.
An "operation" is used here to denote one addition and one multiplication. A further reduction in computational requirements is shown in case $L^m$ and $d^m$ are real. As with Zohar's algorithm, the specialized algorithm requires that all principal minors of $L^m$ be nonzero.