THE AUSTRALIAN NATIONAL UNIVERSITY

TR-CS-97-14

Numerical solution of the eigenvalue problem for Hermitian Toeplitz-like matrices

Michael K. Ng and William F. Trench

July 1997

Joint Computer Science Technical Report Series

Department of Computer Science
Faculty of Engineering and Information Technology

Computer Sciences Laboratory
Research School of Information Sciences and Engineering

This technical report series is published jointly by the Department of Computer Science, Faculty of Engineering and Information Technology, and the Computer Sciences Laboratory, Research School of Information Sciences and Engineering, The Australian National University.

Please direct correspondence regarding this series to:

Technical Reports
Department of Computer Science
Faculty of Engineering and Information Technology
The Australian National University
Canberra ACT 0200
Australia

or send email to:

`Technical.Reports@cs.anu.edu.au`

A list of technical reports, including some abstracts and copies of some full reports may be found at:

`http://cs.anu.edu.au/techreports/`

Recent reports in this series:

TR-CS-97-13 Michael K. Ng. *Blind channel identification and the eigenvalue problem of structured matrices*. July 1997.

TR-CS-97-12 Michael K. Ng. *Preconditioning of elliptic problems by approximation in the transform domain*. July 1997.

TR-CS-97-11 Richard P. Brent, Richard E. Crandall, and Karl Dilcher. *Two new factors of Fermat numbers*. May 1997.

TR-CS-97-10 Andrew Tridgell, Richard Brent, and Brendan McKay. *Parallel integer sorting*. May 1997.

TR-CS-97-09 M. Manzur Murshed and Richard P. Brent. *Constant time algorithms for computing the contour of maximal elements on the Reconfigurable Mesh*. May 1997.

TR-CS-97-08 Xun Qu, Jeffrey Xu Yu, and Richard P. Brent. *A mobile TCP socket*. April 1997.

NUMERICAL SOLUTION OF THE EIGENVALUE PROBLEM FOR HERMITIAN TOEPLITZ–LIKE MATRICES

MICHAEL K. NG * AND WILLIAM F. TRENCH †

Abstract. An iterative method based on displacement structure is proposed for computing eigenvalues and eigenvectors of a class of Hermitian Toeplitz–like matrices which includes matrices of the form T^*T where T is arbitrary Toeplitz matrix, Toeplitz–block matrices and block–Toeplitz matrices. The method obtains a specific individual eigenvalue (i.e., the i-th smallest, where i is a specified integer in $[1, 2, \ldots, n]$) of an $n \times n$ matrix at a computational cost of $O(n^2)$ operations. An associated eigenvector is obtained as a byproduct. The method is more efficient than general purpose methods such as the QR algorithm for obtaining a small number (compared to n) of eigenvalues. Moreover, since the computation of each eigenvalue is independent of the computation of all other eigenvalues, the method is highly parallelizable. Numerical results illustrate the effectiveness of the method.

Key words. Toeplitz matrix, displacement structure, Toeplitz–like matrix, eigenvalue, eigenvector, root–finding

AMS subject classifications. 15A18, 15A57, 65F15

* ACSys, Computer Sciences Laboratory, Research School of Information Sciences and Engineering, The Australian National University, Canberra, ACT 0200, Australia. E-mail: mng@cslab.anu.edu.au.

† Department of Mathematics, Trinity University, 715 Stadium Drive, San Antonio, Texas, 78212, USA. E-mail: wtrench@trinity.edu. Research supported by National Science Foundation Grant DMS-9305856.

1. Introduction. In this paper we consider the eigenvalue problem for an $n \times n$ Hermitian matrix A_n which has displacement structure in the sense that

$$A_n Z_n - Z_n A_n = G_n H_n^T, \tag{1}$$

where Z_n is the shift matrix

$$Z_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix},$$

G_n and H_n are in $\mathbf{C}^{n\times\alpha}$, and α is small compared to n. (For discussions of other types of displacement structure see [8, 10, 11]). The smallest integer α for which (1) holds with some G_n and H_n in $\mathbf{C}^{n\times\alpha}$, is called *the* $\{Z_n, Z_n\}$*-displacement rank of* A_n; we will call it simply the *displacement rank of* A_n.

Henceforth we will say that a matrix which satisfies (1) with α small compared to n is a *Toeplitz–like* matrix. There are many efficient direct methods that exploit displacement structure to invert Toeplitz–like matrices, or to solve Toeplitz–like systems $A_n x = b$ [6, 8, 11]. There are also preconditioned conjugate gradient methods for solving Toeplitz-like systems with $O(n \log n)$ operations [2, 4]. However, numerical solution of the Toeplitz eigenvalue problem has only recently received attention [5, 9, 15, 16]. In particular, Cybenko and Van Loan [5] presented a method for using Levinson's algorithm [12] to find the smallest eigenvalue of an $n \times n$ Hermitian Toeplitz matrix with $O(n^2)$ operations. In [15, 16], Trench extended their method and gave an iterative method for computing arbitrary eigenvalues and associated eigenvectors of Hermitian Toeplitz and Toeplitz–plus–Hankel matrices at a cost of $O(n^2)$ per eigenvalue. The purpose of this paper is to use Trench's method to compute the eigenvalues and eigenvectors of Hermitian Toeplitz–like matrices.

In §2 we propose an algorithm for finding individual eigenvalues of an $n\times n$ matrix with displacement rank not greater than α at a computation cost of $O(\alpha n^2)$ each. In §3 we give examples of Hermitian matrices with displacement structure (1), along with specific formulas for the associated matrices G_n and H_n. In §4 we discuss an application to signal processing. In §5 we describe the results of numerical experiments with the algorithm.

2. The algorithm. The following theorem from [16] provides the motivation and the theoretical basis for the method. Part of this theorem goes back at least to Wilkinson [17]. (For the statement concerning the inertia of $A_n - \lambda I_n$, see also Browne [1]).

THEOREM 2.1. *Let* $A_n = [a_{ij}]_{i,j=1}^n$ *be a Hermitian matrix, and define*

$$A_m = [a_{ij}]_{i,j=1}^m, \quad 1 \le m \le n.$$

Let $p_0(\lambda) = 1$,

$$p_m(\lambda) = \det\,(A_m - \lambda I_m), \quad 1 \le m \le n,$$

and

$$q_m(\lambda) = \frac{p_m(\lambda)}{p_{m-1}(\lambda)}, \quad 1 \le m \le n.$$

Define

$$v_m = \begin{bmatrix} a_{1,m+1} \\ a_{2,m+1} \\ \vdots \\ a_{m,m+1} \end{bmatrix}, \quad 1 \le m \le n-1.$$

Let S_m *be the spectrum of* A_m *and* $\mathcal{S}_n = \cup_{m=1}^{n-1} S_m$. *If* λ *is real let* $\mathrm{Neg}_n(\lambda)$ *be the number (counting multiplicities) of eigenvalues of* A_n *less than* λ. *For each* $\lambda \notin \mathcal{S}_n$ *let* $w_0(\lambda) = 0$ *and*

$$w_m(\lambda) = \begin{bmatrix} w_{1m}(\lambda) \\ w_{2m}(\lambda) \\ \vdots \\ w_{mm}(\lambda) \end{bmatrix}, \quad 1 \le m \le n-1,$$

be the solutions of the systems

$$(A_m - \lambda I_m) w_m(\lambda) = v_m, \quad 1 \le m \le n-1. \tag{2}$$

Define

$$y_m(\lambda) = \begin{bmatrix} w_{m-1}(\lambda) \\ -1 \end{bmatrix}, \quad 2 \le m \le n. \tag{3}$$

Then

$$(A_m - \lambda I_m) y_m(\lambda) = -q_m(\lambda) e_m, \quad 2 \le m \le n, \tag{4}$$

where $e_m = [0\ 0\ \cdots\ 1]^T$ *is the last column of* I_m. *Moreover,*

$$q_m(\lambda) = a_{mm} - \lambda - v_{m-1}^* w_{m-1}(\lambda), \quad 1 \le m \le n,$$

$$q_m'(\lambda) = -1 - \|w_{m-1}(\lambda)\|_2^2,$$

and $\mathrm{Neg}_n(\lambda)$ *equals the number of negative quantities in* $\{q_1(\lambda), q_2(\lambda), \dots, q_n(\lambda)\}$. *Finally, if* λ *is an eigenvalue of* A_n, *then* $y_n(\lambda)$ *is an associated eigenvector.*

Theorem 2.1 provides a way to compute $p_n(\lambda)/p_{n-1}(\lambda)$ and the inertia of $A_n - \lambda I_n$. Therefore, in principle it can be used in conjunction with a root–finding procedure to determine a given eigenvalue λ_i of A_n, provided that λ_i is not "too close" to an eigenvalue of one of the principal submatrices $A_1, A_2, \dots, A_{n-1}$ of A_n. This method is not practical for general Hermitian matrices, because in general $O(n^3)$ operations are required to solve the systems (2) for each value of λ. However, Theorem 2.1 can be useful if A_n is structured so that this computational cost is $O(n^2)$. In [15], Trench described a computational strategy combining Theorem 2.1 with bisection and the Pegasus root–finding method for computing individual eigenvalues and eigenvectors of Hermitian Toeplitz matrices with $O(n^2)$ operations. In [16] he applied the same strategy to Hermitian Toeplitz–plus–Hankel matrices. We will now show that a similar approach can be used to compute individual eigenvalues and eigenvectors of Toeplitz–like matrices.

Henceforth G_m and H_m $(1 \le m \le n)$ are the $m\times\alpha$ matrices obtained by dropping rows $m+1,\dots,n$ from G_n and H_n; thus

$$G_m = U_{mn}G_n \quad \text{and} \quad H_m = U_{mn}H_n, \tag{5}$$

where U_{mn} is the $m\times n$ matrix obtained by dropping the same rows from I_m. We denote the jth column of G_m by

$$g_j^{(m)} = \begin{bmatrix} g_{1j} \\ \vdots \\ g_{mj} \end{bmatrix};$$

thus,

$$G_m = [g_1^{(m)}\ g_2^{(m)} \cdots g_\alpha^{(m)}].$$

The following result of Heinig and Rost [8, p.161] is crucial to our approach.

Lemma 2.2. *If A_n satisfies* (1) *then*

$$A_mZ_m - Z_mA_m = G_mH_m^T - v_me_m^T, \quad 1 \le m \le n-1. \tag{6}$$

Proof. It is easily verified that

$$U_{mn}A_nZ_nU_{mn}^T = A_mZ_m + v_me_m^T \quad \text{and} \quad U_{mn}Z_nA_nU_{mn}^T = Z_mA_m.$$

Therefore we can obtain (6) by multiplying (1) on the left by U_{mn} and on the right by U_{mn}^T, and invoking (5). □

The following algorithm provides an $O(\alpha n^2)$ method for solving the linear systems (2) if A_n satisfies (1) with $G_n, H_n \in \mathbf{C}^{n\times\alpha}$. The algorithm is an adaptation of a recursion formula given in [8, p.161] for solving systems with Toeplitz–like matrices.

Algorithm 2.3. *If $\lambda \notin \mathcal{S}_n$ then $q_1(\lambda),\dots,q_n(\lambda)$ can be computed as follows:*

$$q_1(\lambda) = a_{11} - \lambda, \quad w_1(\lambda) = \frac{a_{12}}{q_1(\lambda)} \quad ,$$

$$f_j^{(1)}(\lambda) = \frac{g_{1j}}{q_1(\lambda)}, \quad 1 \le j \le \alpha,$$

and for $2 \le m \le n$,

$$q_m(\lambda) = a_{mm} - \lambda - v_{m-1}^* w_{m-1}(\lambda), \tag{7}$$

$$y_m(\lambda) = \begin{bmatrix} w_{m-1}(\lambda) \\ -1 \end{bmatrix},$$

$$f_j^{(m)}(\lambda) = \begin{bmatrix} f_{j-1}^{(m-1)}(\lambda) \\ 0 \end{bmatrix} - \frac{(g_{mj} - v_{m-1}^* f_{j-1}^{(m-1)}(\lambda))}{q_m(\lambda)} y_m(\lambda), \quad 1 \le j \le \alpha, \tag{8}$$

and

$$w_m(\lambda) = \begin{bmatrix} 0 \\ w_{m-1}(\lambda) \end{bmatrix} - \left[f_1^{(m)}(\lambda)\ f_2^{(m)}(\lambda) \cdots f_\alpha^{(m)}(\lambda)\right] H_m^T y_m(\lambda). \tag{9}$$

Proof. Adding and subtracting λZ_m on the left side of (6) yields

$$(A_m - \lambda I_m)Z_m - Z_m(A_m - \lambda I_m) = G_m H_m^T - v_m e_m^T, \quad 1 \le m \le n-1. \tag{10}$$

From (3) and (4), for $2 \le m \le n$,

$$e_m^T y_m(\lambda) = -1, \quad Z_m(A_m - \lambda I_m)y_m(\lambda) = 0, \quad \text{and} \quad Z_m(\lambda)y_m(\lambda) = \begin{bmatrix} 0 \\ w_{m-1}(\lambda) \end{bmatrix}.$$

Therefore, multiplying (10) on the right by $y_m(\lambda)$ yields

$$(A_m - \lambda I_m)\begin{bmatrix} 0 \\ w_{m-1}(\lambda) \end{bmatrix} = G_m H_m^T y_m(\lambda) + v_m, \quad 2 \le m \le n.$$

Multiplying by $(A_m - \lambda I_m)^{-1}$ and recalling (2) shows that this is equivalent to

$$w_m(\lambda) = \begin{bmatrix} 0 \\ w_{m-1}(\lambda) \end{bmatrix} - F_m(\lambda)H_m^T y_m(\lambda),$$

where

$$F_m(\lambda) = (A_m - \lambda I_m)^{-1} G_m,$$

which we write in terms of its columns as

$$F_m(\lambda) = [f_1^{(m)}(\lambda)\ f_2^{(m)}(\lambda) \cdots f_\alpha^{(m)}(\lambda)].$$

These columns are the solutions of

$$(A_m - \lambda I_m) f_j^{(m)}(\lambda) = g_j^{(m)} = \begin{bmatrix} g_j^{(m-1)} \\ g_{mj} \end{bmatrix}, \quad 1 \le j \le \alpha. \tag{11}$$

Since

$$(A_{m-1} - \lambda I_{m-1}) f_j^{(m-1)}(\lambda) = g_j^{(m-1)} \quad \text{and} \quad (A_m - \lambda I_m) y_m(\lambda) = -q_m(\lambda) e_m,$$

it follows that the solutions of (11) are given by (8). ☐

3. Examples. The following are examples of Hermitian matrices with the kind of displacement structure indicated in (1).

(i) A Hermitian Toeplitz matrix $T_n = [t_{i-j}]_{i,j=1}^n$ (where $t_{-r} = \bar{t}_r$) has displacement rank at most 2, since

$$T_n Z_n - Z_n T_n = \begin{bmatrix} 1 & 0 \\ 0 & \bar{t}_{n-1} \\ \vdots & \vdots \\ 0 & \bar{t}_1 \end{bmatrix} \begin{bmatrix} \bar{t}_1 & \cdots & \bar{t}_{n-1} & 0 \\ 0 & \cdots & 0 & -1 \end{bmatrix}.$$

(ii) If $T_n = [t_{i-j}]_{i,j=1}^n$ is an arbitrary (not necessarily Hermitian) $n \times n$ Toeplitz matrix, then $A_n = T_n^* T_n$ has displacement rank at most 4 (see [8, p.146] and [10]). It can be shown that A_n satisfies (1) with

$$G_n = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \bar{t}_{-1} & \bar{t}_{n-1} & 0 & b_1 \\ \vdots & \vdots & \vdots & \vdots \\ \bar{t}_{-n+1} & \bar{t}_1 & 0 & b_{n-1} \end{bmatrix} \quad \text{and} \quad H_n = \begin{bmatrix} t_{-1} & -t_{n-1} & c_1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ t_{-n+1} & -t_1 & c_{n-1} & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix},$$

where

$$b_i = \sum_{k=1}^{n} \bar{t}_{k-i+1} t_{k-n} \quad \text{and} \quad c_j = \sum_{k=1}^{n-1} \bar{t}_{k-1} t_{k-j-1}.$$

(iii) A matrix of the form $\sum_{i=1}^{k} T_n^{(i)*} T_n^{(i)}$, where $T_n^{(1)}, \ldots, T_n^{(k)}$ are arbitrary Toeplitz matrices, has displacement rank at most $4k$ [3]. Matrices like this arise in solving the normal equations of Toeplitz least squares problems in signal and image processing [2].

(iv) A Hermitian Toeplitz–block matrix of the form

$$A_n = \begin{bmatrix} T_m^{(1,1)} & T_m^{(1,2)} & \cdots & T_m^{(1,s)} \\ T_m^{(1,2)*} & T_m^{(2,2)} & \cdots & T_m^{(2,s)} \\ \vdots & \vdots & \ddots & \vdots \\ T_m^{(1,s-1)*} & T_m^{(2,s-1)*} & \cdots & T_m^{(s-1,s)} \\ T_m^{(1,s)*} & T_m^{(2,s)*} & \cdots & T_m^{(s,s)} \end{bmatrix}, \tag{12}$$

where $n = sm$ and $\{T_m^{(i,j)}\}_{i,j=1}^{s}$ are Toeplitz matrices given by $[T_m^{(i,j)}]_{k,l=1}^{m} = t_{k-l}^{(i,j)}$, has displacement rank at most $2s$ [8, p.147]. For example, if $s = 2$ it can be shown that (1) holds with $n = 2m$,

$$G_n = \begin{bmatrix} 1 & 0 & t_0^{(1,2)} & 0 \\ 0 & 0 & t_1^{(1,2)} - t_{-m+1}^{(1,1)} & -t_{-m+1}^{(1,2)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & t_{m-1}^{(1,2)} - t_{-1}^{(1,1)} & -t_{-1}^{(1,2)} \\ 0 & 1 & t_0^{(2,2)} & 0 \\ 0 & 0 & t_1^{(2,2)} - t_{-m+1}^{(1,2)} & -t_{-m+1}^{(2,2)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & t_{m-1}^{(2,2)} - t_{-1}^{(1,2)} & -t_{-1}^{(2,2)} \end{bmatrix} \text{ and } H_n = \begin{bmatrix} t_{-1}^{(1,1)} & t_{-1}^{(1,2)} - t_{m-1}^{(1,1)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ t_{-m+1}^{(1,1)} & t_{-m+1}^{(1,2)} - t_1^{(1,1)} & 0 & 0 \\ 0 & -t_0^{(1,1)} & 1 & 0 \\ t_{-1}^{(1,2)} & t_{-1}^{(2,2)} - t_{m-1}^{(1,2)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ t_{-m+1}^{(1,2)} & t_{-m+1}^{(2,2)} - t_1^{(1,2)} & 0 & 0 \\ 0 & -t_0^{(1,2)} & 0 & 1 \end{bmatrix}.$$

(v) For an example closely related to (iv) let B_n be the block–Toeplitz matrix

$$B_n = \left[C_s^{(p-q)}\right]_{p,q=1}^{m},$$

where each block is an $s \times s$ matrix; thus,

$$C_s^{(r)} = \left[c_{ij}^{(r)}\right]_{i,j=1}^{s}.$$

Now let P_n be the $n \times n$ permutation matrix defined as follows: for $k = 1, 2, \ldots, m$, rows $(k-1)s + 1$ through ks of P_n are rows $k, k+m, \ldots, k+(s-1)m$ of I_n. Then $A_n = P_n B_n P_n^T$ is the Toeplitz–block matrix

$$A_n = \left[T_m^{(i,j)}\right]_{i,j=1}^{s},$$

where

$$T_m^{i,j} = \left[c_{ij}^{(p-q)}\right]_{p,q=1}^{m}.$$

Moreover, if B_n is Hermitian then so is A_n; that is, A_n is of the form (12). Finally, if λ is an eigenvalue and x is an associated eigenvector of A_n, then λ is an eigenvalue and $P_n^T x$ is an associated eigenvector of B_n.

4. An application to signal processing. The input $\{x_k\}$ and the output $\{y_k\}$ of a transversal filter of order n are related by

$$y_r = \sum_{k=0}^{n-1} w_k x_{r-k}.$$

In signal processing problems it is often necessary to estimate the filter coefficients $\{w_0, w_1, \ldots, w_{n-1}\}$ given observed values $\{x_1, x_2, \ldots, x_m\}$ and $\{y_1, y_2, \ldots, y_m\}$ of the input and output, where $m > n$. One way to do this is to choose $\{w_0, w_1, \ldots, w_{n-1}\}$ so as to minimize

$$\sigma(w_0, w_1, \ldots, w_{n-1}) = \sum_{r=1}^{m} \left(y_r - \sum_{k=0}^{n-1} w_k x_{r-k} \right)^2,$$

where it is assumed that $x_j = 0$ if $j \leq 0$. An elementary argument shows that $\{w_0, w_1, \ldots, w_{n-1}\}$ should be chosen so that

$$\sum_{j=1}^{n} a_{ij} w_{j-1} = \sum_{r=1}^{m} y_r x_{r-i+1}, \quad 1 \leq i \leq n,$$

where

$$a_{ij} = \sum_{r=1}^{m} x_{r-i+1} x_{r-j+1}.$$

The matrix $A_n = [a_{ij}]_{i,j=1}^{n}$ is given by $A_n = X^T X$, where X is the $m \times n$ Toeplitz matrix

$$X = \begin{bmatrix} x_1 & 0 & \cdots & 0 & 0 \\ x_2 & x_1 & \ddots & & 0 \\ \vdots & x_2 & x_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ x_n & \ddots & \ddots & \ddots & x_1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x_{m-1} & \ddots & \ddots & \ddots & x_{m-n} \\ x_m & x_{m-1} & \cdots & x_{m-n+2} & x_{m-n+1} \end{bmatrix}. \tag{13}$$

The matrix $X^T X$ is called the *normal equations matrix* or the *information matrix* of the corresponding least squares problem [13, 14]. It is an approximation to the the correlation matrix of the input signal data. We are interesting in computing the eigenvalues of $X^T X$ because, for example, the smallest and the largest eigenvalues of $X^T X$ are related to the accuracy of the least squares computations and the stability of least squares algorithms [13, 14]. In [7] it was shown that the filter coefficients that maximize the output signal-to-noise ratio can be obtained from the eigenvector of $X^T X$ associated with its largest eigenvalue.

It can be shown that $A_n = X^T X$ satisfies (1) with

$$G_n = \begin{bmatrix} 0 & 1 & 0 \\ x_m & 0 & u_1 \\ \vdots & \vdots & \vdots \\ x_{m-n+2} & 0 & u_{n-1} \end{bmatrix} \quad \text{and} \quad H_n = \begin{bmatrix} -x_m & v_1 & 0 \\ \vdots & \vdots & \vdots \\ -x_{m-n+2} & v_{n-1} & 0 \\ 0 & 0 & -1 \end{bmatrix},$$

where

$$u_i = \sum_{l=1}^{m-n+1} x_l x_{l+n-i} \quad \text{and} \quad v_j = \sum_{l=j+1}^{m} x_l x_{l-j}.$$

Therefore each iteration of Algorithm 2.3 requires $O(3n^2)$ operations.

5. Numerical results. We tried Algorithm 2.3 on Toeplitz–block matrices (with $s = 2$) as mentioned in §3 and on matrices of the form $T_n^* T_n$ where T_n is an arbitrary real Toeplitz matrix. The elements of these matrices are randomly generated with a uniform distribution in $[-10, 10]$. All computations were done with Matlab in double precision.

Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of a Toeplitz-like matrix A_n, and suppose we wish to find λ_i, where i is a specified integer in $[1, \ldots, n]$. We assume that λ_i is not an eigenvalue of any of the principal submatrices $A_1, \ldots, A_{n-1}$. We first find an interval (α, β) containing λ_i but not any other eigenvalues of A_n, or any eigenvalues of A_{n-1}. On such an interval q_n is continuous. In [15] it was shown that α and β satisfy this requirement if and only if

$$\mathrm{Neg}_n(\alpha) = i - 1, \quad \mathrm{Neg}_n(\beta) = i,$$

$$q_n(\alpha) > 0, \quad \text{and} \quad q_n(\beta) < 0,$$

and a strategy was given for obtaining (α, β) by means of bisection. After (α, β) is determined, we use the Matlab M-file "fzero" to find λ_i as a root of the function $q_n(\lambda)$. (This root–finding algorithm was originated by T. Dekker and further improved by R. Brent; see Matlab on-line documentation.) We stop the iteration for λ_i when the difference between successive iterates μ_{k-1} and μ_k obtained by the root finder satisfies the inequality

$$|\mu_k - \mu_{k-1}| \leq 4 \times 10^{-11} \times \max\{|\mu_k|, 1\}.$$

To check the accuracy of the individual eigenvalues and associated eigenvectors of the randomly generated Toeplitz–like matrices, we computed the residual norms

$$\sigma_i = \frac{\|A_n y_n(\tilde{\lambda}_i) - \tilde{\lambda}_i y_n(\tilde{\lambda}_i)\|_2}{\|y_n(\tilde{\lambda}_i)\|_2},$$

where $\tilde{\lambda}_i$ is the approximate ith eigenvalue and $y_n(\tilde{\lambda}_i)$ (as defined in (3) with $\lambda = \tilde{\lambda}_i$) is an approximate λ_i–eigenvector. Tables 1 and 2 show the distribution of $\{\sigma_i\}$ for 50 randomly generated matrices of order 100, 50 of order 500, and 50 of order 1000, for two types of Toeplitz–like matrices. Table 3 lists the average number of iterations per eigenvalue for two types of Toeplitz–like matrices.

For each randomly generated Toeplitz–like matrix of order n we formed the diagonal matrix D_n consisting of the computed eigenvalues and the matrix Ω_n whose columns are the corresponding computed eigenvectors. For each matrix we computed the reconstruction and orthogonality errors

$$\tau = \frac{\|A_n - \Omega_n D_n \Omega_n^T\|_F}{\|A_n\|_F} \quad \text{and} \quad \nu = \frac{\|I_n - \Omega_n \Omega_n^T\|_F}{\sqrt{n}},$$

where $\|\cdot\|_F$ is the Frobenius norm. The results are shown in Tables 4 and 5.

We also tried Algorithm 2.3 on matrices of the form $X^T X$ where X is as in (13), with $m = 1024$ and $n = 128$. We considered 50 cases with $\{x_1, \ldots, x_{1024}\}$ generated by the second-order autoregressive (AR) process

$$x_k - 1.4x_{k-1} + 0.5x_{k-2} = \phi_k,$$

and 50 cases with $\{x_1, \ldots, x_{1024}\}$ generated by the second-order moving-average (MA) process

$$x_k = \phi_k + 0.75\phi_{k-1} + 0.25\phi_{k-2}.$$

In all instances $\{\phi_k\}$ is a Gaussian process with mean zero and variance one, and $E(\phi_j \phi_k) = \delta_{jk}$. Tables 6 and 7 show the distribution of the residual norm σ_i and the relative error between the eigenvalues computed by Algorithm 2.3 and those computed by the QR method, respectively. Table 8 shows the values of τ and ν for these two input processes. The average numbers of iterations per eigenvalue for the AR and MA processes were 10.23 and 10.54 respectively.

TABLE 1
Distribution of errors $\{\sigma_i\}$ for 50 matrices $A_n = T_n^* T_n$, where T_n are randomly generated nonsymmetric $n \times n$ Toeplitz matrices.

Interval	Number of errors $n = 100$	$n = 500$	$n = 1000$
$[10^{-2}, 10^{-1})$	0	0	0
$[10^{-3}, 10^{-2})$	0	0	1
$[10^{-4}, 10^{-3})$	0	1	2
$[10^{-5}, 10^{-4})$	1	10	33
$[10^{-6}, 10^{-5})$	5	177	306
$[10^{-7}, 10^{-6})$	20	259	1848
$[10^{-8}, 10^{-7})$	945	8591	21646
$[10^{-9}, 10^{-8})$	2951	14467	24742
$[10^{-10}, 10^{-9})$	923	1345	1343
$[10^{-11}, 10^{-10})$	113	94	56
$[10^{-12}, 10^{-11})$	42	56	23
$[10^{-13}, 10^{-12})$	0	0	0

TABLE 2
Distribution of errors $\{\sigma_i\}$ for 50 randomly generated Toeplitz-block matrices with $s = 2$ and $n = 2m$.

Interval	Number of errors		
	$n = 100$	$n = 500$	$n = 1000$
$[10^{-2}, 10^{-1})$	0	0	0
$[10^{-3}, 10^{-2})$	0	0	0
$[10^{-4}, 10^{-3})$	0	0	1
$[10^{-5}, 10^{-4})$	0	14	15
$[10^{-6}, 10^{-5})$	1	101	136
$[10^{-7}, 10^{-6})$	4	391	692
$[10^{-8}, 10^{-7})$	27	3478	9758
$[10^{-9}, 10^{-8})$	158	10961	20091
$[10^{-10}, 10^{-9})$	2807	8659	17949
$[10^{-11}, 10^{-10})$	1709	1267	1234
$[10^{-12}, 10^{-11})$	262	112	101
$[10^{-13}, 10^{-12})$	32	17	23

TABLE 3
Average number of iterations per eigenvalue for computations summarized in Tables 1 and 2.

Type	Number of iterations		
	$n = 100$	$n = 500$	$n = 1000$
$T_n^* T_n$ where T_n are nonsymmetric Toeplitz matrices	10.12	10.18	11.26
Toeplitz–block matrices	10.34	10.59	11.09

TABLE 4
Reconstruction and orthogonality errors for 50 matrices $A_n = T_n^* T_n$ where T_n are randomly generated nonsymmetric Toeplitz matrices.

	$n = 100$		$n = 500$		$n = 1000$	
Interval	τ	μ	τ	μ	τ	μ
$[10^{-5}, 10^{-4})$	0	0	1	1	1	1
$[10^{-6}, 10^{-5})$	0	0	1	1	2	2
$[10^{-7}, 10^{-6})$	0	0	3	3	11	10
$[10^{-8}, 10^{-7})$	1	2	12	13	29	31
$[10^{-9}, 10^{-8})$	17	13	27	28	7	6
$[10^{-10}, 10^{-9})$	28	32	6	4	0	0
$[10^{-11}, 10^{-10})$	4	3	0	0	0	0

TABLE 5
Reconstruction and orthogonality errors for 50 randomly generated Toeplitz-block matrices with $s = 2$ and $n = 2m$.

	$n = 100$		$n = 500$		$n = 1000$	
Interval	τ	μ	τ	μ	τ	μ
$[10^{-7}, 10^{-6})$	0	0	0	1	8	7
$[10^{-8}, 10^{-7})$	1	1	2	3	13	23
$[10^{-9}, 10^{-8})$	2	3	25	15	21	18
$[10^{-10}, 10^{-9})$	22	21	17	26	8	2
$[10^{-11}, 10^{-10})$	22	24	6	5	0	0
$[10^{-12}, 10^{-11})$	3	1	0	0	0	0

TABLE 6

Distribution of errors $\{\sigma_i\}$ for 50 matrices $X^T X$ with $m = 1024$ and $n = 128$.

Interval	Number of errors	
	AR Process	MA Process
$[10^{-7}, 10^{-6})$	2	1
$[10^{-8}, 10^{-7})$	28	31
$[10^{-9}, 10^{-8})$	1657	1824
$[10^{-10}, 10^{-9})$	3899	3657
$[10^{-11}, 10^{-10})$	814	887

TABLE 7

Distribution of the relative error between the eigenvalues computed by Algorithm 2.3 method and those computed by QR method for 50 matrices $X^T X$ with $m = 1024$ and $n = 128$.

Interval	Number of errors	
	AR Process	MA Process
$[10^{-7}, 10^{-6})$	3	4
$[10^{-8}, 10^{-7})$	136	71
$[10^{-9}, 10^{-8})$	1959	2356
$[10^{-10}, 10^{-9})$	3736	3612
$[10^{-11}, 10^{-10})$	566	357

TABLE 8

Reconstruction and orthogonality errors for 50 matrices for $X^T X$ with $m = 1024$ and $n = 128$.

	AR Process		MA Process	
Interval	τ	μ	τ	μ
$[10^{-8}, 10^{-7})$	4	3	3	4
$[10^{-9}, 10^{-8})$	19	20	17	19
$[10^{-10}, 10^{-9})$	26	25	28	26
$[10^{-11}, 10^{-10})$	1	2	2	1

6. Summary. The experimental results reported here show that Algorithm 2.3 is an efficient and effective method for computing individual eigenvalues of Hermitian Toeplitz-like matrices. For an $n \times n$ Toeplitz–like matrix, the computational cost of each eigenvalue and an associated eigenvector is $O(n^2)$ operations. The method is more efficient than general purpose methods such as the QR algorithm for obtaining a small number (compared to n) of eigenvalues. (See [15]). Since the computation of each eigenvalue is independent of the computation of all others, the method is highly parallelizable. Moreover, if $q_1(\lambda)$, ..., $q_n(\lambda)$ are computed with a parallel processing machine utilizing as many processors as necessary to exploit the full parallelism in the algorithm, the multiplications as well as additions required to compute in (7), (8) and (9) can be carried out simultaneously. The inner products in (7), (8) and (9) can also be computed simultaneously by employing parallel processors in $O(\log n)$ time units. Therefore, the computations of $\{q_1(\lambda),\ldots,q_n(\lambda)\}$ when performed by $O(n)$ parallel processors, can be accomplished in $O(n \log n)$ time units. Hence the computations of each eigenvalue, when performed by $O(n)$ processors, can be accomplished in $O(n \log n)$ time.

REFERENCES

[1] E. T. Browne, *On the separation property of the roots of the secular equation*, Amer. J. Math., 52 (1930), pp. 841–850.

[2] R. Chan, J. Nagy, and R. Plemmons, *Circulant Preconditioned Toeplitz Least Squares Iterations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 80–97.

[3] R. Chan, J. Nagy, and R. Plemmons, *Displacement Preconditioner for Toeplitz Least Squares Iterations*, Elec. Trans. Numer. Anal., 2 (1994), pp. 44–56.

[4] R. Chan and M. Ng, *Conjugate Gradient Methods for Solving Toeplitz Systems*, SIAM Review, 38 (1996), pp. 427–482.

[5] G. Cybenko and C. Van Loan, *Computing the Minimum Eigenvalue of a Symmetric Positive Definite Toeplitz Matrix*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 123–131.

[6] I. Gohberg, T. Kailath and V. Olshevsky, *Fast Gaussian Elimination with Partial Pivoting for Matrices with Displacement Structure*, Math. Comp., 64 (1995), pp. 1557–1576.

[7] S. Haykin, *Adaptive Filter Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1991.

[8] G. Heinig and K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Operator Theory: Advances and Applications V13, Birkhäuser Verlag, Stuttgart, 1984.

[9] Y. H. Hu and S. Y. Kung, *Toeplitz Eigensystem Solver*, IEEE Trans. Acoustics, Speech, Sig. Proc., 33 (1985), pp. 1264–1271.

[10] T. Kailath, S. Kung and M. Morf, *Displacement Ranks of Matrices and Linear Equations*, J. Math. Anal. Appl., 68 (1979), pp. 395–407.

[11] T. Kailath and A. Sayed, *Displacement Structure: Theory and Applications*, SIAM Review, 37 (1995), pp. 297–386.

[12] N. Levinson, *The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction*, J. Math. and Phys., 25 (1946), pp. 261–278.

[13] W. Ferng, G. Golub and R. Plemmons, *Adaptive Lanczos Methods for Recursive Condition Estimation*, Numer. Algo., 1 (1991), pp. 1–19.

[14] D. Pierce and R. Plemmons, *Tracking the Condition Number for RLS in Signal Processing*, Math. Control Signals Systems, 5 (1992), pp. 23–39.

[15] W. F. Trench, *Numerical Solution of the Eigenvalue Problem for Hermitian Toeplitz Matrices*, SIAM J. Matrix Th. Appl., 9 (1988), pp. 291–303.

[16] W. F. Trench, *Numerical Solution of the Eigenvalue Problem for Efficiently Structured Hermitian Matrices*, Linear Algebra Appl., 154–156 (1991), pp. 415–432.

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