NUMERICAL SOLUTION OF THE EIGENVALUE PROBLEM FOR HERMITIAN TOEPLITZ MATRICES

William F. Trench*

SIAM J. Matrix Anal. Appl. 10 (1989) 135-156

Abstract. An iterative procedure is proposed for computing the eigenvalues and eigenvectors of Hermitian Toeplitz matrices. The computational cost per eigenvalue–eigenvector for a matrix of order n is $0(n^2)$ in serial mode. Results of numerical experiments on Kac–Murdock–Szegö matrices and randomly generated real symmetric Toeplitz matrices of orders 100, 150, 300, 500, and 1000 are included.

I. Introduction. Here we present a method for computing the eigenvalues and eigenvectors of Hermitian Toeplitz matrices; i.e., matrices of the form

$$T_n = (t_{i-j})_{i,j=1}^n$$

with $t_r = \bar{t}_{-r}$. The method rests specifically and crucially on the special structure of T_n. There are efficient algorithms which exploit this simple structure to invert such matrices, or to solve systems $T_nX = Y$. There is also an extensive literature on the asymptotic distribution of the eigenvalues of a family $\{T_n\}$ of Hermitian Toeplitz matrices as $n \to \infty$, in the case where the $\{t_m\}$ are the Fourier coefficients of a function f which satisfies suitable integrability conditions. However, the development of efficient methods designed specifically to compute the eigenvalues and eigenvectors of these matrices is still in its early stages.

Several recent papers ([5], [7], [9], [18], [23]) have dealt with the spectral structure of Hermitian Toeplitz matrices, and numerical methods aimed mainly at finding the smallest eigenvalue of a positive definite Hermitian Toeplitz matrix have appeared ([8], [11], [14], [15]). Some of these use inverse iteration with Rayleigh quotient shifting, exploiting the Levinson algorithm [17] for solving Toeplitz systems. The author [22] has proposed a method which, on the basis of preliminary numerical experiments, appears to provide an effective procedure for computing the eigenvalues of Hermitian Toeplitz matrices generated by rational functions, at a cost per eigenvalue which is essentially independent of the order of the matrix. (Autocorrelation matrices of ARMA processes are of this kind.)

The method presented here combines the Levinson–Durbin algorithm [6] for the shifted matrices $T_m - \lambda I_m$ $(1 \leq m \leq n-1)$ with an iterative root finding procedure to locate the zeros of the rational function

$$q_n(\lambda) = p_n(\lambda)/p_{n-1}(\lambda), \tag{1}$$

where

$$p_m(\lambda) = \det[T_m - \lambda I_m],\ 1 \leq m \leq n. \tag{2}$$

The basic idea of this approach is not original with us. Cybenko and Van Loan [8] used the Levinson–Durbin algorithm and Newton's method to compute the smallest eigenvalue of a symmetric positive definite Toeplitz matrix, and our work should be considered to be a continuation of theirs. However, our method will determine any eigenvalue of T_n which is not also an eigenvalue of any of the nested submatrices $T_1, \ldots, T_{n-1}$ (an assumption also required by Cybenko and Van Loan). The corresponding eigenvectors are obtained as byproducts.

Delsarte and Genin [9] have used arguments based on the Levinson–Durbin algorithm as applied to the shifted matrices $T_m - \lambda I_m$ to obtain theoretical results concerning the spectra of Hermitian Toeplitz matrices. For a result related to their work, see also Wilkes and Hayes [23].

2. The Theoretical Basis for the Method.

Most of the results in this section are not new, although we believe that this presentation in specific reference to the eigenvalue problem is somewhat more explicit and complete than previous discussions. In any case, it seems appropriate to include it here for the reader's convenience.

Since the eigenvalues of T_n are real, we assume throughout that λ is real.

Let

$$U_{n-1} = [t_1, t_2, \ldots, t_{n-1}]^T \qquad (^T = \text{ transpose}).$$

If λ is not an eigenvalue of T_{n-1}, then let

$$X_{n-1}(\lambda) = [x_{1,n-1}(\lambda), \ldots, x_{n-1,n-1}(\lambda)]^T$$

be the solution of

$$(T_{n-1} - \lambda I_{n-1}) X_{n-1}(\lambda) = U_{n-1}, \tag{3}$$

and define

$$Y_n(\lambda) = \begin{bmatrix} -1 \\ X_{n-1}(\lambda) \end{bmatrix}. \tag{4}$$

Recall the definitions (1) and (2) of $q_n(\lambda)$ and $p_m(\lambda)$. In the following, $\| \quad \|$ is the Euclidean norm.

THEOREM 1. *If λ is not an eigenvalue of T_{n-1}, then*

$$q_n(\lambda) = t_0 - \lambda - \bar{U}_{n-1}^T X_{n-1}(\lambda) \tag{5}$$

and

$$q_n'(\lambda) = -1 - \|X_{n-1}(\lambda)\|^2. \tag{6}$$

If, in addition, λ is an eigenvalue of T_n, then $Y_n(\lambda)$ is an associated eigenvector.

PROOF: We partition $T_n - \lambda I_n$ in the form

$$T_n - \lambda I_n = \begin{bmatrix} t_0 - \lambda & \bar{U}_{n-1}^T \\ U_{n-1} & T_{n-1} - \lambda I_{n-1} \end{bmatrix}. \tag{7}$$

Subtract $x_{j,n-1}(\lambda)$ times column $j+1$ from the first column of (7) for $j = 1, \ldots, n-1$, and invoke (3) to obtain

$$p_n(\lambda) = \begin{vmatrix} t_0 - \lambda - \bar{U}_{n-1}^T X_{n-1}(\lambda) & \bar{U}_{n-1}^T \\ 0 & T_{n-1} - \lambda I_{n-1} \end{vmatrix}$$

$$= (t_0 - \lambda - \bar{U}_{n-1}^T X_{n-1}(\lambda)) p_{n-1}(\lambda),$$

which implies (5). From (3), (4), (5), and (7),

$$(T_n - \lambda I_n) Y_n(\lambda) = -q_n(\lambda)[1, 0, \ldots, 0]^T; \tag{8}$$

hence, if λ is an eigenvalue of T_n, then $Y_n(\lambda)$ is an associated eigenvector.

To verify (6), we differentiate (5):

$$\begin{aligned} q_n'(\lambda) &= -1 - \bar{U}_{n-1}^T X_{n-1}'(\lambda) \\ &= -1 - \bar{X}_{n-1}^T(\lambda)(T_{n-1} - \lambda I_{n-1}) X_{n-1}'(\lambda), \end{aligned} \tag{9}$$

where the second equality follows from (3) and the Hermitian symmetry of $T_{n-1} - \lambda I_{n-1}$. Since differentiating (3) shows that

$$(T_{n-1} - \lambda I_{n-1})X'_{n-1}(\lambda) = X_{n-1}(\lambda),$$

(9) implies (6).

The formula (6) is due to Cybenko and Van Loan [8]; however, they did not explicitly identify $q_n(\lambda)$ as the ratio $p_n(\lambda)/p_{n-1}(\lambda)$.

Except for a missing minus sign on the right, (3) is the Yule–Walker equation for $T_{n-1} - \lambda I_{n-1}$ (cf. [6]). The following theorem is essentially a statement of the Levinson–Durbin algorithm for solving (3), with minor changes to account for the fact that the diagonal element of the matrix in (3) is $t_0 - \lambda$ rather than unity. We omit the proof.

THEOREM 2. *If $T_m - \lambda I_m$ is nonsingular for $1 \le m \le n-1$, then (3) can be solved recursively as follows: Let*

$$x_{11}(\lambda) = t_1/(t_0 - \lambda), \qquad \Delta_1(\lambda) = t_0 - \lambda, \tag{10}$$

and, for $2 \le m \le n-1$,

$$\Delta_m(\lambda) = [1 - |x_{m-1,m-1}(\lambda)|^2]\Delta_{m-1}(\lambda), \tag{11}$$

$$x_{mm}(\lambda) = \Delta_m^{-1}(\lambda)[t_m - \sum_{j=1}^{m-1} t_{m-j}x_{j,m-1}(\lambda)], \tag{12}$$

and

$$x_{jm}(\lambda) = x_{j,m-1}(\lambda) - x_{mm}(\lambda)\bar{x}_{m-j,m-1}(\lambda),\ 1 \le j \le m-1. \tag{13}$$

For convenience, we say that a real number λ is *nondefective with respect to* T_n if it is not an eigenvalue of any of the principal submatrices $T_1, \ldots, T_{n-1}$. Conversely, λ is *defective with respect to* T_n if it is an eigenvalue of any of these matrices. An eigenvalue of T_n which is not simultaneously an eigenvalue of any of the principal submatrices will be said to be a *nondefective eigenvalue* of T_n. From the Cauchy separation theorem, a nondefective eigenvalue must be of multiplicity one. (Note that these are nonstandard usages of *defective* and *nondefective*.)

Cybenko [6] has shown that if $m \geq 2$ and

$$L_m(\lambda) = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ -x_{1,m-1}(\lambda) & 1 & \ldots & 0 & 0 \\ -x_{2,m-1}(\lambda) & -x_{1,m-2}(\lambda) & \ldots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -x_{m-1,m-1}(\lambda) & -x_{m-2,m-2}(\lambda) & \ldots & -x_{1,1}(\lambda) & 1 \end{bmatrix},$$

then

$$\bar{L}_m^T(\lambda)(T_m - \lambda I_m)L_m(\lambda) = diag[\Delta_m(\lambda), \ldots, \Delta_1(\lambda)]. \tag{14}$$

Because of Sylvester's law of inertia, this implies the following theorem, which has been used previously in the algorithms of Cybenko and Van Loan [8] and Hu and Kung [15] for computing the smallest eigenvalue of a positive definite Hermitian Toeplitz matrix, and is also crucial for the more general algorithm presented here.

THEOREM 3: *Let* $Neg_m(\lambda)$ *be the number of eigenvalues of* T_m *(counting multiplicities) less than* λ. *Then* $Neg_m(\lambda)$ *equals the number of negative values among* $\{\Delta_1(\lambda), \ldots, \Delta_m(\lambda)\}$, *provided that* λ *is nondefective with respect to* T_m.

Since $\det L_m(\lambda) = 1$, (14) implies that

$$p_m(\lambda) = \prod_{j=1}^{m} \Delta_m(\lambda), \ 1 \leq m \leq n, \tag{15}$$

which is essentially equivalent to a formula obtained in [21] for the determinant of a Hermitian Toeplitz matrix. Setting $m = n$ in (15) shows that $p_n(\lambda) = \Delta_n(\lambda)p_{n-1}(\lambda)$; hence

$$q_n(\lambda) = \Delta_n(\lambda). \tag{16}$$

Henceforth we will use $q_n(\lambda)$ and $\Delta_n(\lambda)$ interchangeably.

Notice that it is not necessary to carry out the computations in (13) for $m = n - 1$ in order to compute $q_n(\lambda)$ from (16), as it would be if we wished to use the formula (5) obtained earlier. (However, (16) requires that $T_m - \lambda I_m$ be nonsingular for $1 \leq m \leq n - 1$, while (5) requires only that $T_{n-1} - \lambda I_{n-1}$ be nonsingular.)

THEOREM 4: *Suppose that α and β are nondefective with respect to T_n, and that (α, β) contains exactly one eigenvalue (with multiplicity one) of T_n. Suppose also that neither α nor β is an eigenvalue of T_n. Then (α, β) contains no eigenvalues of T_{n-1} if and only if $\Delta_n(\alpha) > 0$ and $\Delta_n(\beta) < 0$.*

PROOF: Since $Neg_n(\beta) = 1 + Neg_n(\alpha)$ by assumption, Theorem 3 implies that the set $\{\Delta_1(\beta), \ldots, \Delta_n(\beta)\}$ has exactly one more negative member than the set $\{\Delta_1(\alpha), \ldots, \Delta_n(\alpha)\}$. Therefore, if either $\Delta_n(\alpha) < 0$ or $\Delta_n(\beta) > 0$, the set $\{\Delta_1(\beta), \ldots, \Delta_{n-1}(\beta)\}$ must contain more negative members than the set $\{\Delta_1(\alpha), \ldots, \Delta_{n-1}(\alpha)\}$, and therefore (α, β) contains at least one eigenvalue of T_{n-1}, by Theorem 3. On the other hand, if $\Delta_n(\alpha) > 0$ and $\Delta_n(\beta) < 0$, then the two sets mentioned

in the last sentence must contain the same number of negative elements, and Theorem 3 implies that $Neg_{n-1}(\beta) = Neg_{n-1}(\alpha)$; i.e., that T_{n-1} has no eigenvalues in (α, β).

As observed in [8], The idea of computing $p_n(\lambda)/p_{n-1}(\lambda)$ by partitioning a Hermitian matrix as in (7) and then locating its zeros by combining inertia computations with a root finding method has been used by other authors (see, e.g., [19] and [24]); however, this approach requires $O(n^3)$ operations for the general Hermitian matrix, rather than the $O(n^2)$ required for Toeplitz matrices.

In connection with his work on real centrosymmetric matrices, Andrews [1] defined a vector $V = [v_1, \ldots, v_n]^T$ to be *symmetric* if

$$v_j = v_{n-j+1}, \ 1 \le j \le n, \tag{17}$$

or *skew-symmetric* if

$$v_j = -v_{n-j+1}, \ 1 \le j \le n. \tag{18}$$

Cantoni and Butler [5] have shown that if T is a real symmetric Toeplitz matrix of order n then R has an orthonormal basis consisting of $n - [n/2]$ symmetric and $[n/2]$ skew-symmetric eigenvectors of T. (Here $[x]$ is the integer part of x.) For convenience, let us say that an eigenvalue of T_n is *even* or *odd* if it has an associated eigenvector which satisfies (17) or (18), respectively. It is clear from (11) with $m = n - 1$ that λ is a nondefective eigenvalue of a real symmetric Toeplitz matrix if and only if $x_{n-1,n-1}(\lambda) = \pm 1$. From the form of the associated eigenvector $Y_n(\lambda)$ in (4), we can see more specifically that λ is a nondefective even eigenvalue of T_n if and only if $x_{n-1,n-1}(\lambda) = -1$, or a nondefective odd eigenvalue if and only if $x_{n-1,n-1}(\lambda) = 1$.

3. The Iterative Procedure.

If λ is defective with respect to T_n, then $q_n(\lambda)$ cannot be computed by means of Theorem 3. For practical purposes it is more appropriate to observe that $q_n(\lambda)$ cannot be computed in this way if at least one of the quantities $\Delta_1(\lambda), \ldots, \Delta_{n-1}(\lambda)$ is so small as to cause overflow in (12) for some m in $\{1, \ldots, n-1\}$. We will discuss this further in Section 5; however, for now it suffices to say that in the numerical experiments reported in Section 4, which comprise the computation of thousands of eigenvalues, there was not a single instance in which computation was terminated for this reason. Therefore, we will assume in this section that the eigenvalues of T_n (or at least those that we are trying to compute) are nondefective, and that none of the approximants to the eigenvalues generated by the procedure that we are about to describe are sufficiently close to being defective so as to cause overflow in (12).

We use an iterative procedure to locate the eigenvalues of T_n as the zeros of $q_n = p_n/p_{n-1}$. The iteration terminates when the difference between successive iterates is sufficiently small. In the following description of the procedure, we assume that $\Delta_n(\lambda) \neq 0$ for every value of λ encountered during the iteration. This is for convenience only; obviously, if $\Delta_n(\lambda)$ "underflows" to zero, then λ is an acceptable approximation to an eigenvalue. (This did not occur in any of our computations.)

Let the eigenvalues of T_n be

$$\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n,$$

and suppose that we wish to find a single nondefective eigenvalue λ_i. Our first task is to find an interval (α, β) which contains λ_i, but does not contain any other eigenvalues of T_n or any eigenvalues of T_{n-1}. On such an interval $q_n(\lambda)$ is continuous. Obviously α and β satisfy the first requirement if and only if

$$\text{Neg}_n(\alpha) = i - 1 \quad \text{and} \quad \text{Neg}_n(\beta) = i, \tag{19}$$

and, given this, Theorem 4 implies that the second holds if and only if

$$\Delta_n(\alpha) > 0 \quad \text{and} \quad \Delta_n(\beta) < 0. \tag{20}$$

In the following, $\mathrm{Neg}_n(\lambda)$ is computed by means of Theorem 3.

To start, we find α and β, by trial and error, such that

$$\mathrm{Neg}_n(\alpha) \leq i-1 \quad \text{and} \quad \mathrm{Neg}_n(\beta) \geq i. \tag{21}$$

If (19) and (20) hold for this α and β, then this phase of the computation is finished. If not, let $\gamma = (\alpha+\beta)/2$. If $\mathrm{Neg}_n(\gamma) \leq i-1$, replace α by γ; if $\mathrm{Neg}_n(\gamma) \geq i$, replace β by γ. Repeat this until (19) and (20) both hold, which must occur after finitely many steps.

Since q_n is continuous on the interval (α, β) that we have just determined, we can now switch from bisection to a more efficient zero finding method to locate λ_i. The method of false position [20] was unacceptably slow, but the Pegasus modification of this method yielded consistently good results. Since this procedure is well described in the literature (see, e.g. [10], [20]), we will not describe it here, except to say that if $\{\mu_j\}$ is the sequence of iterates produced by the Pegasus computation, starting with $\mu_0 = \alpha$ and $\mu_1 = \beta$, then we terminate this phase of the computation at the first integer r such that

$$|\mu_r - \mu_{r-1}| \ < \ .5(1+\mu_r)10^{-K}, \tag{22}$$

where K is a suitable positive integer. We then compute $\Delta_n(\mu_r)$ from (10) – (13), and continue (13) with $m = n-1$ (which is not required to compute $\Delta_n(\mu_r)$, as mentioned earlier) to compute $x_{1,n-1}(\mu_r), \ldots, x_{n-2,n-1}(\mu_r)$. Then we use Newton's method to obtain a final approximation to λ_i:

$$\begin{aligned} \mu_{r+1} &= \mu_r - \Delta_n(\mu_r)/\Delta_n'(\mu_r) \\ &= \mu_r + \Delta_n(\mu_r)/(1 + \|X_{n-1}(\mu_r)\|^2) \end{aligned}$$

(cf. (6) and (16)).

This application of Newton's method is "for good measure," and can probably be omitted without great loss. We included it without rigorously evaluating its effect because in some cases it appeared to allow the use of a smaller integer K in (22) without degrading the results, and we report it here since it was used in most of the computations reported in Section 4. We did not use Newton's method as our principal iterative technique (after determining α and β as in (19) and (20)), since it could produce a sequence of iterates which converge to another eigenvalue or does not converge at all. The Pegasus method does not suffer from this defect, and it has a respectable order of convergence (approximately 1.642).

It may be of interest to note that Newton's method as applied to this problem is actually a form of Rayleigh quotient iteration. To see this, suppose that λ is an approximation to an eigenvalue of T_n. Then the vector $Y_n(\lambda)$ in (4) is an approximation to a corresponding eigenvector, and a new approximation $\hat{\lambda}$ to the eigenvalue can be obtained by computing the Rayleigh quotient

$$\hat{\lambda} = \frac{\bar{Y}_n^T(\lambda) T_n Y_n(\lambda)}{\|Y_n\|^2}. \tag{23}$$

However, from (8) and (16),

$$T_n Y_n(\lambda) = \lambda Y_n(\lambda) - \Delta_n(\lambda)[1, 0, \ldots, 0]^T,$$

so that

$$\bar{Y}_n^T(\lambda) T_n Y_n(\lambda) = \lambda \|Y_n(\lambda)\|^2 + \Delta_n(\lambda).$$

Since

$$\|Y_n(\lambda)\|^2 = 1 + \|X_{n-1}(\lambda)\|^2 = -\Delta_n'(\lambda)$$

(cf. (6) and (16)), it now follows that the Rayleigh quotient $\hat{\lambda}$ in (23) can be rewritten as

$$\hat{\lambda} = \lambda - \Delta_n(\lambda)/\Delta_n'(\lambda),$$

which establishes our point.

Now suppose that we wish to find eigenvalues $\lambda_p, \ldots, \lambda_q$, where $1 \le p < q \le n$. Since it would be wasteful to simply apply the procedure just described independently for $i = p, \ldots, q$, we will define a method for finding $\xi_{p-1}, \ldots, \xi_q$ such that

$$\xi_{i-1} < \lambda_i < \xi_i, \ p \le i \le q. \tag{24}$$

Having accomplished this, we then apply the above described procedure for $i = p, \ldots, q$, taking the initial points in the search for λ_i to be $\alpha = \xi_{i-1}$ and $\beta = \xi_i$. (Clearly, (24) implies (21) in this case.) It is to be understood that as each ξ_i is determined, $\Delta_n(\xi_i)$ is retained for subsequent use.

The inequalities (24) are equivalent to

$$\begin{aligned} &\mathrm{Neg}_n(\xi_{p-1}) \le p-1, \\ &\mathrm{Neg}_n(\xi_i) = i, \ p \le i \le q-1, \\ &\mathrm{Neg}_n(\xi_q) \ge q. \end{aligned} \tag{25}$$

We specify the method for choosing $\xi_{p-1}, \ldots, \xi_q$ inductively. We start by choosing a and b, by trial and error, such that $\mathrm{Neg}_n(a) \le p-1$ and $\mathrm{Neg}_n(b) \ge q$, and let

$\xi_{p-1} = a$ and $\xi_q = b$. Now suppose that at some step of our inductive procedure ξ_{p-1} and ξ_q have been specified, but at least one of the intermediate points $\xi_p, \ldots, \xi_{q-1}$ has not. Let r and s be the smallest integers such that $p \le r < s \le q$ and ξ_r has not been selected, while ξ_s has. Define

$$\gamma = (\xi_{r-1} + \xi_s)/2 \tag{26}$$

and $k = \mathrm{Neg}_n(\gamma)$. If $r = p$ and $k < p - 1$ (which can occur only if the inequality holds in (25)), then we replace ξ_{p-1} by γ. Similarly, if $s = q$ and $k > q$, then we replace ξ_q by γ. In all other cases, $r - 1 \le k \le s$, and we let $\xi_k = \gamma$.

This procedure merely replaces a previously selected ξ_k unless k satisfies the stronger inequalities $r \le k \le s - 1$; however, the bisection (26) will obviously cause the selection process to be completed in a finite number of steps.

Since ξ_{i-1} is no longer needed after λ_i has been obtained, λ_i can be stored in the location previously occupied by ξ_{i-1}.

4. Computational Results.

We considered real symmetric matrices only. All computations reported here were performed in double precision (15+ decimal places) in Fortran 77. The computations for all matrices of order less than 1000 were performed on an IBM PC AT. Those for matrices of order 1000 were performed on an IBM PS/2 Model 60. Both machines were equipped with the 80287 coprocessor. Due to the limitations of available computing equipment, we made no attempt to use parallel processing to solve the Levinson–Durbin system (3). Therefore, the computation of each eigenvalue and its associated eigenvector with our implementation of the proposed method requires $0(n^2)$ steps, where the "constant" buried in the "0" depends, of course, on the number of iterations required for the given eigenvalue. Although

this number depends upon the eigenvalue itself and on the starting values (α and β), its average value over all eigenvalue–eigenvector pairs for a matrix of order n appears to be essentially independent of n. Of course, it depends on K in (22). In the computations reported here, we took $K = 10$.

We consider two kinds of matrices: the Kac–Murdock–Szegö (KMS) matrices

$$T_n = (\rho^{|i-j|})_{i,j=1}^{n} , \quad (0 < \rho < 1) \tag{27}$$

which are discussed in [13] and [16], and matrices

$$T_n = (t_{i-j})_{i,j=1}^{n} ,$$

in which the defining elements $t_0, \ldots, t_{n-1}$ were randomly generated with a uniform distribution in $[-10, 10]$.

The eigenvalues of the KMS matrices can be computed quite easily, even on a hand-held calculator. It is shown in [13] that if

$$\sin(n+1)\gamma - 2\rho \sin n\gamma + \rho^2 \sin(n-1)\gamma = 0, \tag{28}$$

then the quantity

$$\lambda = (1-\rho^2)(1 - 2\rho\cos\gamma + \rho^2)^{-1} \tag{29}$$

is an eigenvalue of T_n in (27). Moreover, it is also shown in [13] that (28) has roots $\gamma_1, \ldots, \gamma_n$ which satisfy the inequalities

$$0 < \gamma_1 < \frac{\pi}{n+1} < \gamma_2 < \frac{2\pi}{n+1} < \cdots < \gamma_n < \frac{n\pi}{n+1}.$$

Given such precise information on their locations, it is a simple matter to find $\gamma_1, \ldots, \gamma_n$ by standard root-finding methods, and then to compute the eigenvalues $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ from

$$\lambda_i = (1-\rho^2)(1 - 2\rho\cos\gamma_{n-i+1} + \rho^2)^{-1},\ 1 \le i \le n. \tag{30}$$

For a considerable extension of this idea, see [22].

We used the algorithm proposed here to compute all eigenvalues of KMS matrices of orders n=100, 300, 500, and 1000 for various values of ρ. We also computed the same eigenvalues by the "exact" method; that is, by solving (28) iteratively with the Pegasus procedure to obtain $\gamma_1, \ldots, \gamma_n$, and then computing $\lambda_1, \ldots, \lambda_n$ from (30). We terminated the iteration for each γ_i as soon as the difference between successive iterates was less than 10^{-14}. We then computed the fractional error

$$f_i = (\hat{\lambda}_i - \tilde{\lambda}_i)/\tilde{\lambda}_i, \tag{31}$$

where $\hat{\lambda}_i$ and $\tilde{\lambda}_i$ are the estimates of λ_i obtained from our general algorithm and the "exact" method, respectively. The distributions of these fractional errors are shown in Tables 1, 2, 3 and 4; e.g., Table 1 shows that for $n = 100$ and $\rho = .5$, 14 of the fractional errors were in the interval $[10^{-14}, 10^{-13})$.

Tables 5 and 6 summarize results obtained in computing all eigenvalues of 20 randomly generated matrices of order 100, 24 of order 150, 22 of order 300, 5 of

order 500, and 2 of order 1000. As mentioned above, the eigenvectors were obtained as byproducts. We attempted to assess the results as follows.

We computed

$$Q_i = |q_n(\hat{\lambda}_i)| \,, \quad 1 \le i \le n \tag{32}$$

(or, equivalently, $Q_i = |\Delta_n(\hat{\lambda}_i)|$), where $\hat{\lambda}_i$ is the final estimate of λ_i. We also computed

$$R_i = \min\{|x_{n-1,n-1}(\hat{\lambda}_i) - 1| \,,\ |x_{n-1,n-1}(\lambda) + 1|\} \tag{33}$$

It is obvious from (1) that $Q_i = 0$ if $\hat{\lambda}_i = \lambda_i$. Also, since λ_i is an eigenvalue of T_n if $x_{n-1,n-1}(\lambda_i) = \pm 1$, $R_i = 0$ if $\hat{\lambda}_i = \lambda_i$. Table 5 shows the percentage distributions of $\{Q_i\}$ and $\{R_i\}$. Here n is the order of the matrix and m is the number of matrices of that order for which the results are given. Under each value of n there are two columns, headed Q and R, which show the percentage distributions of $\{Q_i\}$ and $\{R_i\}$, respectively, for all m matrices of the given order n. For example, 34.58 percent of the $\{Q_i\}$ and 11.74 percent of the $\{R_i\}$ fell in the interval $[10^{-9}, 10^{-8})$ for n=300.

After a considerable portion of the computations summarized in Table 5 had been completed, we decided that a more decisive measure of error should be calculated for the randomly generated matrices; namely,

$$\sigma_i = \|T_n - \hat{\lambda}_i Y_n(\hat{\lambda}_i)\| \ / \ \|Y_n(\hat{\lambda}_i)\|, \tag{34}$$

since $Y_n(\hat{\lambda}_i)$ (as defined in (4) with $\lambda = \hat{\lambda}_i$) is an approximate λ_i-eigenvector. Table 6 shows the percentage distribution of $\{\sigma_i\}$ for a subclass of the matrices considered in Table 5; again, m is the number of matrices of the given order n which are included in this subclass.

The computations in (10)–(13) require approximately n^2 flops for each λ. With $K = 10$ in (22), these computations were performed on the average approximately eleven times per eigenvalue (and this was essentially independent of the particular matrix or its order). Let us extrapolate from these computations, and assume that this method requires approximately $M(K)n^2$ flops per eigenvalue, where $M(10) \approx 11$. By comparison, standard QR requires approximately $2n^3/3$ flops for the preliminary tridiagonalization of T_n, after which all the the eigenvalues can be computed with $O(n)$ flops [12, Section 8.2]. On the basis of this count only, it would seem that the method presented here has a clear advantage over standard QR if it is desired to compute N eigenvalues ($1 \leq N \leq n$) of T_n, provided that N is small compared to $(2n)/3M(K)$, while the advantage shifts to standard QR if this is not so. In the context of parallel processing, the determination of the crossover point is more complicated; since the computations for distinct eigenvalues are completely independent of each other with the present method, it is straightforward to distribute the labor of computing many eigenvectors among multiple processors. Moreover, the memory requirement for the present method is $O(n)$, compared to $O(n^2)$ for standard QR.

The approximate average running times required on the IBM PC AT to find all eigenvalues and eigenvectors of T_n, with $K = 10$ in (22) and without computing σ_i (cf.(34)), were 24 minutes, 81 minutes, 10.6 hours, and 49 hours for n=100, 150, 300, and 500, respectively. For those runs in which σ_i was computed, the average running times were approximately 27 minutes, 88 minutes, 11.6 hours, and

54 hours, respectively. For the matrices of order 1000, the average running time per eigenvalue-eigenvector pair was approximately 15 minutes on the PS/2 Model 60.

5. The Effects of Defectiveness

Our program included a command to terminate computation of $q_n(\lambda) = \Delta_n(\lambda)$ if

$$|1 - x^2_{m-1,m-1}(\lambda)| < 10^{-J} \tag{35}$$

for some m in $\{1, \ldots, n-1\}$. The purpose of this test is to prevent overflow in (12) if λ is too close to an eigenvalue of one of the principal submatrices $T_1, \ldots, T_{n-1}$ of T_n. In the computations reported in Section 4 we took $J = 9$ (recall that $K = 10$ in (22)), and termination for this reason never occurred. Thus, the practical effect of defectiveness is not that it is likely to cause overflow (although this can be forced to happen in contrived situations); rather, it effects the accuracy of the results.

In most cases where the error indicators f_i, Q_i, and R_i, were relatively large, we were able to ascertain that the eigenvalues in question were close to being defective. For example, it can be seen in Table 5 that Q_i was in the interval [1,10) for one of the 6600 eigenvalues computed for randomly generated matrices of order 300. This was $\hat{\lambda}$=122.418638510399, with $q_{300}(\hat{\lambda}) \cong 8.45$. To test for defectiveness, we reduced J in (35) to 4 and attempted to compute $q_{300}(\hat{\lambda})$. The calculation terminated with $m = 298$. Subsequent calculation showed that $q_{298}(\hat{\lambda}) \cong .31 \times 10^{-5}$, indicating that $\hat{\lambda}$ was close to an eigenvalue of T_{298}. Examination of other cases in which the error indicators were unusually large yielded similar results.

The results in Table 1 for the KMS matrix with $\rho = .5$ and $n = 100$ show that the fractional error f_i for one eigenvalue is in the interval $[10^{-8}, 10^{-7})$, while

all the others are less than 10^{-12}. The eigenvalue for which this occurred is $\lambda = 1$, which is defective; indeed, it is straightforward to verify that if $\rho = .5$ and $n = 3m + 1 \quad (m = 0, 1, 2, \ldots)$, then $\gamma = \pi/3$ satisfies (28) and therefore, from (29), $\lambda = 1$ is an eigenvalue of T_n. Hence, $\lambda = 1$ is an eigenvalue of 33 principal submatrices of T_{100}.

Although our results indicate that defectiveness in the sense that we have defined it is not a major problem for the matrices that we have considered, it would still be worthwhile to develop methods to overcome it. Clearly, defectiveness is a problem — theoretically — mainly because the Levinson–Durbin algorithm for solving the Yule–Walker equation (3) requires that all the principal submatrices of $T_n - \lambda I_n$ be nonsingular; i.e., that $T_n - \lambda I_n$ be "strongly nonsingular." Alternative methods have been proposed for solving Toeplitz systems with matrices which are not strongly nonsingular; for discussions of such methods, see [2], [3] and [4]. A possible direction for future research would be to incorporate some of the ideas in these references into the present method.

6. Acknowledgment

The author thanks G. Cybenko and the referees for several suggestions for improving the manuscript.

References

[1] A.L. Andrew, *Eigenvectors of certain matrices,* Lin. Alg. Appl., 7 (1973), 151–162.

[2] R.P. Brent, F.G. Gustavson, and D.Y.Y. Yun, *Fast solution of Toeplitz systems of equations and computation of Padé approximants,* Jnl. of Algorithms 1, (1980), 259–295.

[3] J.R. Bunch, *Stability of methods for solving Toeplitz systems of equations,* SIAM J. Sci. Stat. Comput. 6 (1985), 349–364.

[4] _____, *The weak and strong stability of algorithms in numerical linear algebra,* Linear Alg. and Its Appl. 88/89 (1987), 49–66.

[5] A. Cantoni and F. Butler, *Eigenvalues and eigenvectors of symmetric centrosymmetric matrices,* Lin. Alg. Appl., 13 (1976), pp. 275–288.

[6] G. Cybenko, *The numerical stability of the Levinson–Durbin algorithm for Toeplitz systems of equations,* SIAM J. Sci. Stat. Comput., 1 (1980), pp. 303–319.

[7] _____, *On the eigenstructure of Toeplitz matrices,* IEEE Trans. Acoustics, Speech, Sig. Proc., Vol ASSP-32 (1984), pp. 275–288.

[8] 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.

[9] P. Delsarte and Y. Genin, *Spectral properties of finite Toeplitz matrices,* in *Mathematical Theory of Networks and Systems,* Proc. MTNS–83 International Symposium, Beer Sheva, Israel (1983), pp. 194–213.

[10] M. Dowell and P. Jaratt, *The "Pegasus" method for computing the root of an equation,* BIT 12 (1972), pp. 503–508.

[11] D.R. Fuhrmann and B. Liu, *Approximating the eigenvalues of a symmetric Toeplitz matrix,* Proc. of the 21st Annual Allerton Conference on Communications, Control and Computing (1983), pp. 1046–1055.

[12] E. H. Golub and C. Van Loan, *Matrix Computations,* Johns Hopkins Uni-

versity Press, 1983.

[13] U. Grenander and G. Szegö, *Toeplitz Forms and Their Applications,* University of California Press, 1958.

[14] Y.H. Hu and S.Y. Kung, *Highly concurrent Toeplitz eigensystem solver for high resolution spectral estimation,* Proc. ICASSP 83 (1983), pp. 1422–1425.

[15] _____, *Toeplitz eigensystem solver,* IEEE Trans. Acoustics, Speech, Sig. Proc., Vol ASSP–33 (1985), pp. 1264–1271.

[16] M. Kac, W.L. Murdock, and G. Szegö, *On the eigenvalues of certain Hermitian forms,* J. Rat. Mech. and Anal., 2 (1953), pp. 767–800.

[17] N. Levinson, *The Weiner RMS (root mean square) error criterion in filter design and prediction,* J. Math. Phys., 25 (1947), pp. 261–278.

[18] J. Makhoul, *On the eigenvectors of symmetric Toeplitz matrices,* IEEE Trans. Acoustics, Speech, Sig. Proc., Vol ASSP–29 (1981), pp. 868–872.

[19] B. N. Parlett, *The Symmetric Eigenvalue Problem,* Prentice–Hall, 1980.

[20] A. Ralston and P. Rabinowitz, *A First Course in Numerical Analysis* (2nd ed.), McGraw–Hill, 1978.

[21] W. F. Trench, *An algorithm for the inversion of finite Toeplitz matrices,* SIAM J. Appl. Math., 12 (1964), pp. 515–522.

[22] _____, *Numerical solution of the eigenvalue problem for symmetric rationally generated Toeplitz matrices,* SIAM J. Matrix Th. Appl. , 9 (1988), pp. 291–303.

[23] D.M. Wilkes and M.H. Hayes, *An eigenvalue recursion for Toeplitz ma-*

trices, IEEE Tran. Acoustics, Speech, Sig. Proc., Vol. ASSP–35 (1987), pp. 907–909.

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