A Fast Algorithm for Computing the Smallest Eigenvalue of a Symmetric Positive Definite Toeplitz Matrix

Nicola Mastronardi *Marc Van Barel*
Raf Vandebril

Report TW 493, May 2007

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

A Fast Algorithm for Computing the Smallest Eigenvalue of a Symmetric Positive Definite Toeplitz Matrix

Nicola Mastronardi *Marc Van Barel*
Raf Vandebril

Report TW 493, May 2007

Department of Computer Science, K.U.Leuven

Abstract

Recent progress in signal processing and estimation has generated considerable interest in the problem of computing the smallest eigenvalue of a symmetric positive definite Toeplitz matrix. Several algorithms have been proposed in the literature. Many of them compute the smallest eigenvalue in an iterative fashion, relying on the Levinson–Durbin solution of sequences of Yule–Walker systems.

Exploiting the properties of two algorithms recently developed for estimating a lower and an upper bound of the smallest singular value of upper triangular matrices, respectively, an algorithm for computing bounds to the smallest eigenvalue of a symmetric positive definite Toeplitz matrix has been recently derived [20]. The algorithm relies on the computation of the R factor of the QR–factorization of the Toeplitz matrix and the inverse of R. The simultaneous computation of R and R^{-1} is efficiently accomplished by the generalized Schur algorithm. Unfortunately, due to the weak stability of the generalized Schur algorithm, only a rough approximation of the smallest eigenvalue can be computed. In this paper, exploiting the properties of the latter algorithm, a numerical method to compute the smallest eigenvalue and the corresponding eigenvector of symmetric positive definite Toeplitz matrices in an accurate way is proposed.

Keywords : Toeplitz matrix, symmetric positive definite matrix, generalized Schur algorithm, eigenvalues, incremental norm estimation, signal processing.
AMS(MOS) Classification : Primary : 65F05, Secondary : 65F15, 65F35.

A FAST ALGORITHM FOR COMPUTING THE SMALLEST EIGENVALUE OF A SYMMETRIC POSITIVE DEFINITE TOEPLITZ MATRIX*

N. MASTRONARDI† ‡, M. VAN BAREL‡, AND R. VANDEBRIL‡

Abstract. Recent progress in signal processing and estimation has generated considerable interest in the problem of computing the smallest eigenvalue of a symmetric positive definite Toeplitz matrix. Several algorithms have been proposed in the literature. Many of them compute the smallest eigenvalue in an iterative fashion, relying on the Levinson–Durbin solution of sequences of Yule–Walker systems.

Exploiting the properties of two algorithms recently developed for estimating a lower and an upper bound of the smallest singular value of upper triangular matrices, respectively, an algorithm for computing bounds to the smallest eigenvalue of a symmetric positive definite Toeplitz matrix has been recently derived [20]. The algorithm relies on the computation of the R factor of the QR–factorization of the Toeplitz matrix and the inverse of R. The simultaneous computation of R and R^{-1} is efficiently accomplished by the generalized Schur algorithm. Unfortunately, due to the weak stability of the generalized Schur algorithm, only a rough approximation of the smallest eigenvalue can be computed. In this paper, exploiting the properties of the latter algorithm, a numerical method to compute the smallest eigenvalue and the corresponding eigenvector of symmetric positive definite Toeplitz matrices in an accurate way is proposed.

AMS subject classifications. 65F05, 65F15, 65F35.

Key words. Toeplitz matrix, symmetric positive definite matrix, generalized Schur algorithm, eigenvalues, incremental norm estimation, signal processing.

1. Introduction.

The problem of computing the smallest eigenvalue of a symmetric positive definite (SPD) Toeplitz matrix and the corresponding eigenvector is of considerable interest in signal processing and estimation applications.

Given the covariance sequence of observed data, Pisarenko [26] suggested a method to determine the sinusoidal frequencies from the eigenvector associated to the smallest eigenvalue of the covariance matrix, a symmetric positive definite Toeplitz matrix T.

Some results on the localization of the smallest eigenvalue of T are presented in [23, 24].

Many algorithms have been proposed in the literature to compute the smallest eigenvalue of T [4, 5, 10, 11, 13, 14, 15, 16, 17, 18, 21, 22, 23, 24, 25, 28, 29, 30, 31].

Given $A \in \mathbb{R}^{m\times n},\ m \gg n,\ b \in \mathbb{R}^m$, and given a lower bound σ of the smallest singular value of A, a way to compute a lower bound of the smallest singular value $\hat{\sigma}$ of the augmented matrix $[A|b]$ is proposed in [6]. The latter lower bound is, quite

*The research of the first author was partially supported by MIUR, grant number 2004015437. The research of the last two authors was partially supported by the Research Council K.U.Leuven, project OT/05/40 (Large rank structured matrix computations), CoE EF/05/006 Optimization in Engineering (OPTEC), by the Fund for Scientific Research–Flanders (Belgium), G.0455.0 (RHPH: Riemann-Hilbert problems, random matrices and Padé-Hermite approximation), G.0423.05 (RAM: Rational modelling: optimal conditioning and stable algorithms), and by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Belgian Network DYSCO (Dynamical Systems, Control, and Optimization). The third author has a grant as "Postdoctoraal Onderzoeker" from the Fund for Scientific Research–Flanders (Belgium). The scientific responsibility rests with the authors.

†Istituto per le Applicazioni del Calcolo "M. Picone", sede di Bari, Consiglio Nazionale delle Ricerche, Via G. Amendola, 122/I, I-70126 Bari, Italy. email:n.mastronardi@ba.iac.cnr.it

‡Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium. email: {Raf.Vandebril, Marc.VanBarel, Nicola.Mastronardi}@cs.kuleuven.be

often, a good approximation of the smallest singular value of the augmented matrix. Recursively repeating the procedure, an algorithm for computing a lower bound of the smallest singular value of a SPD Toeplitz matrix is proposed in [20]. The algorithm relies on the simultaneous computation of the matrix R of the QR–factorization of the Toeplitz matrix and its inverse R^{-1} by means of the generalized Schur algorithm (GSA).

The generalized Schur algorithm is a fast algorithm for computing such kind of factorizations for matrices having small displacement rank [12]. It relies on the knowledge of the so called generators of the matrix rather than the matrix itself.

In this paper, taking the algorithm described in [20] into account and exploiting the properties of the generators of the matrix, we propose a fast algorithm for computing the smallest eigenvalue and corresponding eigenvector of symmetric positive definite Toeplitz matrices in an accurate way.

The paper is organized as follows. In § 2 the algorithm described in [6] for computing the lower bound of the smallest singular value of the augmented matrix $[A|b]$ is shortly introduced. The particular version of GSA is described in § 3. An extensive version of the latter three paragraphs can be found in [20]. We include them here to make the paper self–contained. The proposed algorithm is described in § 4 followed by the numerical examples, reported in § 5, and by the conclusions.

2. Computing a lower bound for the smallest singular value. A lower bound for the smallest singular value $\sigma(\hat{A})$ of the augmented matrix $\hat{A} \equiv [A|b]$, $A \in \mathbb{R}^{m\times n}$, $m \gg n$, $b \in \mathbb{R}^m$ is derived in [6], known a lower bound of the smallest singular value $\sigma(A)$ of A. It relies on the following theorem.

THEOREM 2.1. *[6]. Let $\delta > 0$, $\delta \in \mathbb{R}$. Let $A \in \mathbb{R}^{m\times n}$, $m \gg n$, be a full–rank matrix and let $b \in \mathbb{R}^m$. Let $\sigma(A)$ and $\sigma(\hat{A})$ be the smallest positive singular values of A and $\hat{A} \equiv [A|b]$, respectively and $\sigma(A) > \delta > 0$. Let*

$$\hat{x} = \arg\min_x \|Ax - b\|_2 \quad \text{and} \quad \rho = b - A\hat{x}.$$

Then

$$\begin{array}{ll} \sigma(\hat{A}) > \delta & \text{if } \|\rho\|_2 = 0, \\ \sigma(\hat{A}) > \min\{\delta, \|\rho\|_2\} & \text{if } \|\rho\|_2 > 0 \text{ and } \|\hat{x}\|_2 = 0, \\ \sigma(\hat{A}) > \delta\sqrt{E(\delta, \hat{x}, \rho)} & \text{if } \|\rho\|_2 > 0 \text{ and } \|\hat{x}\|_2 > 0, \end{array}$$

with

$$E(\delta, x, \rho) = 1 + \frac{1}{2}\left(B(\delta, x, \rho) - \sqrt{B^2(\delta, x, \rho) + 4\|x\|_2^2}\right)$$

and

$$B(\delta, x, \rho) = \|x\|_2^2 + \frac{\|\rho\|_2^2}{\delta^2} - 1.$$

□

Based on this theorem, an algorithm* for estimating a lower bound of the smallest singular value of a full–rank matrix can be easily derived.

*The algorithms in this paper are written in a `matlab`–like style. `Matlab` is a registered trademark of The MathWorks, Inc.

%ALGORITHM 1. Lower bound of the smallest singular value of a full–rank matrix A.
%INPUT: R, the R factor of the QR–factorization of A.
%OUTPUT: δ, a lower bound of the smallest singular value of A.
`function`$[\delta] =$`fassino`(R);

$\delta = |R(1,1)|$;
`for` $k = 1 : n-1$,
  $X(1:k, k+1) = R^{-1}(1:k, 1:k)R(1:k, k+1)$;
  $\rho = |R(k+1, k+1)|$;
  $\delta = \delta\sqrt{E(\delta, X(1:k, k+1), \rho)}$;
`end`

The algorithm requires to compute the $n \times n$ upper matrix of the R factor of the QR–factorization of A. This is accomplished with standard techniques, e.g., see [9], in $2n^2(m - n/3)$ floating point operations. Moreover, at step k, the linear system $R(1:k, 1:k)X(1:k, k+1) = R(1:k, k+1)$ needs to be solved, which requires $O(k^2)$ operations. Therefore the complexity of Algorithm 1 is $O(mn^2 + n^3)$. The following lemma emphasizes the relationship between R and the strictly upper triangular matrix X.

LEMMA 2.1. *[20]. Let R be a nonsingular upper triangular matrix and X be the strictly upper triangular matrix computed by Algorithm 1 with input the matrix R. Then*

$$X(1:k, k+1) = -R(k+1, k+1)R^{-1}(1:k, k+1), \quad k = 1, 2, \ldots, n-1. \tag{2.1}$$

The strictly upper triangular part of the matrix X, i.e., the sequence of vectors computed in Algorithm 1, are the columns of the strictly upper triangular part of the inverse of R, scaled by the entries in the main diagonal, respectively.

Hence, the computational complexity of Algorithm 1 can be reduced if the inverse of R can be computed in a fast way and given as input to an adapted version of Algorithm 1.

%ALGORITHM 2. Lower bound of the smallest singular value of a full–rank matrix A.
%INPUT: R, R^{-1}, with R the R factor of the QR–factorization of A.
%OUTPUT: δ, a lower bound of the smallest singular value of A.
`function`$[\delta] =$`fassino_adapted`(R, R^{-1});

$\delta = |R(1,1)|$;
`for` $k = 1 : n-1$,
  $X(1:k, k+1) = -R(k+1, k+1)R^{-1}(1:k, k+1)$;
  $\rho = |R^(k+1, k+1)|$;
  $\delta = \delta\sqrt{E(\delta, X(1:k, k+1), \rho)}$;
`end`

We observe that only the diagonal entries of the matrix R are involved in Algorithm 2. These entries can be also computed as the reciprocal of the diagonal entries of R^{-1}.

In Section 3 we show that the R factor of the QR–factorization of a nonsingular Toeplitz matrix T and the inverse of R can be computed with $O(n^2)$ computational complexity by means of the generalized Schur algorithm.

3. Generalized Schur Algorithm. Let

$$T = \begin{bmatrix} t_1 & t_2 & \ddots & t_n \\ t_2 & t_1 & \ddots & \ddots \\ \ddots & \ddots & \ddots & t_2 \\ t_n & \ddots & t_2 & t_1 \end{bmatrix}. \tag{3.1}$$

Define

$$M = \left[\begin{array}{c|c} T^T T & I_n \\ \hline I_n & 0_n \end{array}\right]. \tag{3.2}$$

The R factor of the QR–factorization of T and its inverse R^{-1} can be retrieved from the LDL^T factorization, with L lower triangular and D diagonal matrices, respectively, of the following block–structured matrix,

$$M = \left[\begin{array}{c|c} T^T T & I_n \\ \hline I_n & 0_n \end{array}\right] = LDL^T,$$

with I_n and 0_n the identity matrix and the null matrix of order n, respectively. In fact, it can be easily shown that

$$M = LDL^T = \left[\begin{array}{c|c} R^T & \\ \hline R^{-1} & R^{-1} \end{array}\right] \left[\begin{array}{c|c} I_n & \\ \hline & -I_n \end{array}\right] \left[\begin{array}{c|c} R & R^{-T} \\ \hline & R^{-T} \end{array}\right]. \tag{3.3}$$

Therefore, to compute R and R^{-1} it is sufficient to compute the first n columns of L. This can be accomplished with $O(n^2)$ floating point operations by means of the generalized Schur algorithm (GSA).

In this section we describe how GSA can be used to compute R and its inverse R^{-1}. A comprehensive treatment of the topic can be found in, e.g., [12].

Let $Z \in \mathbb{R}^{n \times n}$ be the shift matrix

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

and

$$\Phi = Z \oplus Z. \tag{3.4}$$

It turns out that

$$M - \Phi M \Phi^T = G \hat{D} G^T,$$

with $\hat{D} = \mathrm{diag}(1, 1, -1, -1)$ and $G \in \mathbb{R}^{2n \times 4}$ called a *generator* matrix.

Denote $v = T^T(T(:,1))$. The columns of G can be chosen as

$$G(:,1) = \frac{1}{\sqrt{v(1)}} \begin{bmatrix} v \\ \hline e_1^{(n)} \end{bmatrix}, \; G(:,2) = \begin{bmatrix} 0 \\ t_2 \\ \vdots \\ t_n \\ \hline 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$G(:,3) = \begin{bmatrix} 0 \\ G(2:2n-1,1) \end{bmatrix}, \; G(:,4) = \begin{bmatrix} 0 \\ t_n \\ \vdots \\ t_2 \\ \hline 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.5}$$

Let $w \in \mathbb{R}^2$. In the sequel, we denote by `giv` the function that computes the parameters $[c_G, s_G]$ of the Givens rotation G:

$$[c_G, s_G] = \texttt{giv}(w_1, w_2) \;\text{ such that }\; \begin{bmatrix} c_G & s_G \\ -s_G & c_G \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \sqrt{w_1^2 + w_2^2} \\ 0 \end{bmatrix}.$$

Moreover, suppose $w_1 > w_2$. We denote by `hyp` the function that computes the parameters $[c_H, s_H]$ of the hyperbolic[†] rotation H,

$$[c_H, s_H] = \texttt{hyp}(w_1, w_2) \;\text{ such that }\; \begin{bmatrix} c_H & -s_H \\ -s_H & c_H \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \sqrt{w_1^2 - w_2^2} \\ 0 \end{bmatrix}.$$

Let `function`$[G] =$ `gener`(T) be the `matlab`–like function with input the SPD Toeplitz T and output the corresponding generator matrix G. Since the number of columns of the generator matrix G is $4 \ll n$, the GSA for computing R and R^{-1} has $O(n^2)$ computational complexity. It can be summarized in the following algorithm.

% ALGORITHM 3. Generalized Schur algorithm.
% INPUT: G, the generator matrix of the Toeplitz matrix T.
% OUTPUT: R and R^{-1}, with R the R factor of a QR–factorization of T.
`function`$[R, R^{-1}, z_1, z_2] =$`schur`(G);

(A.1) `for` $k = 1 : n$,
(A.2) $\quad [c_G, s_G] = \texttt{giv}(G(k,1), G(k,2))$;
(A.3) $\quad G(k : n+k, 1 : 2) = G(k : n+k, 1 : 2) \begin{bmatrix} c_G & s_G \\ -s_G & c_G \end{bmatrix}$;
(A.4) $\quad [c_G, s_G] = \texttt{giv}(G(k,3), G(k,4))$;
(A.5) $\quad G(k : n+k-1, 3 : 4) = G(k : n+k-1, 3 : 4) \begin{bmatrix} c_G & s_G \\ -s_G & c_G \end{bmatrix}$;
(A.6) $\quad$ `if` $i == n$,

[†]Hyperbolic rotations can be computed in different ways. For "stable" implementations see [1, 3].

(A.7) $z_1 = G(n+1:2n,1)$;
(A.8) $z_1 = G(n+1:2n,1)$;
(A.9) `end`
(A.10) $[c_H, s_H] = \mathtt{hyp}(G(k,1), G(k,3))$;
(A.11) $G(k:n+k,[1,3]) = G(k:n+k,[1,3]) \begin{bmatrix} c_H & -s_H \\ -s_H & c_H \end{bmatrix}$;
(A.12) $R(k,k:n) = G(k:n,1)^T$;
(A.13) $R^{-1}(1:k,k) = G(n+1:n+k,1)$;
(A.14) $G(:,1) = \Phi G(:,1)$;
(A.15) `end`

Each iteration of the latter algorithm involves two products of Givens rotations by an $n \times 2$ matrix, each of those can be accomplished with $6n$ floating point operations, followed by the product of an hyperbolic rotation by an $n \times 2$ matrix, accomplished with $6n$ floating point operations. Therefore the computational complexity of GSA is $18n^2$ floating point operations. We remark that GSA exhibits a lot of parallelism that can be exploited to reduce the computational complexity. For instance, the products involving the Givens rotations and the hyperbolic rotations can be easily done in parallel.

We observe that the hyperbolic rotation in (A.10) is defined ($G(k,1) > G(k,3)$) because of the positive definiteness of T. The meaning of the vectors z_1 and z_2 will become clear in the next section.

4. Computation of the smallest eigenvalue of a symmetric positive definite Toeplitz matrix. The generalized Schur algorithm yields the R factor of the QR–factorization of an SPD Toeplitz matrix T and the inverse R^{-1} of the R in an efficient way. Algorithm 2, applied to R and R^{-1}, computes a lower bound for the smallest singular value of R, i.e., the smallest eigenvalue of the SPD matrix T.

Combining Algorithm 2 and 3 in an iterative fashion, an algorithm for computing a strict lower bound of the smallest eigenvalue of an SPD Toeplitz matrix is derived in [20].

```
% ALGORITHM 4. Computation of a lower bound of the smallest eigenvalue
%                of a SPD Toeplitz matrix T.
% INPUT:  T, a SPD Toeplitz matrix of order n,
%         tol, a fixed tolerance,
% OUTPUT:  λn^(−), a lower bound of the smallest eigenvalue of T;
function [λn^(−)] =small_eig_T(T, tol);
λn^(−) = 0;
Iflag = 0;
while Iflag == 0,
      [G] = gener(T);
      (R, R^−1) =schur(G);
      if R(n, n) > tol,
         [λ] =fassino_adapted(R, R^−1);
         λn^(−) = λn^(−) + λ;
         T = T − λI_n;
      else
         Iflag = 1;
      end
```

`end`

Unfortunately, since Algorithms 1 and 2 suffer from a loss of accuracy (see the error analysis in [6]) and because of the weak stability of GSA [27], respectively, the threshold `tol` was chosen equal to $\sqrt{n} \times 10^{-6}$.

Starting from the approximation of the smallest eigenvalue of T yielded by algorithm 4, we now show how to compute the latter eigenvalue in a more accurate way.

Let $T^{(0)} = T$. Define $T^{(i)}$ as the symmetric positive matrix generated at the i–th iteration of the latter algorithm and let $\lambda_j^{(i)},\ j = 1, \ldots, n$ be the eigenvalues of $T^{(i)}$, with

$$\lambda_1^{(i)} \geq \lambda_2^{(i)} \geq \cdots \geq \lambda_{n-1}^{(i)} \geq \lambda_n^{(i)} > 0.$$

Suppose, without loss of generality, $\lambda_{n-1}^{(i)} > \lambda_n^{(i)}$ (see remark 4.1). The following theorem holds.

THEOREM 4.1. *Let $\left\{T^{(i)}\right\}_{i=0}^{\infty}$ be the sequence of SPD Toeplitz matrices generated at each iteration of Algorithm 4, with corresponding eigenvalues*

$$\lambda_1^{(i)} \geq \lambda_2^{(i)} \geq \cdots \geq \lambda_{n-1}^{(i)} > \lambda_n^{(i)} > 0, \tag{4.1}$$

such that

$$\lim_{i \to \infty} \lambda_n^{(i)} = 0.$$

Let $\hat{r}_n^{(i)}$ be the last column of $R_{-1}^{(i)} \equiv {R^{(i)}}^{-1}$, the inverse of the R–factor of the QR–factorization of $T^{(i)}$. Moreover, let

$$T^{(i)} = U^{(i)} \Lambda^{(i)} {U^{(i)}}^T$$

be the eigenvalue decomposition of $T^{(i)}$, with

$$\Lambda^{(i)} = \operatorname{diag}(\lambda_1^{(i)}, \lambda_2^{(i)}, \ldots, \lambda_{n-1}^{(i)}, \lambda_n^{(i)}),$$

and

$$U^{(i)} = [u_1^{(i)}, u_2^{(i)}, \ldots, u_{n-1}^{(i)}, u_n^{(i)}]$$

the orthogonal matrix of the eigenvectors. Then

$$u_n^{(i)} \to \gamma^{(n)} \hat{r}_n^{(i)} \text{ as } \ i \to \infty,$$

with $\gamma^{(n)} \in \mathbb{R}, \gamma^{(n)} \neq 0$.

Proof. Let

$$R^{(i)} \equiv \begin{bmatrix} S^{(i)} & r^{(i)} \\ & \rho^{(i)} \end{bmatrix}, \tag{4.2}$$

with $S^{(i)} \in \mathbb{R}^{(n-1)\times(n-1)}$ upper triangular, $r^{(i)} \in \mathbb{R}^{n-1}$, and $\rho^{(i)} \in \mathbb{R}$. It turns out

$$R_{-1}^{(i)} = \begin{bmatrix} S_{-1}^{(i)} & s_{-1}^{(i)} \\ & \rho_{-1}^{(i)} \end{bmatrix},$$

with $S_{-1}^{(i)} = {S^{(i)}}^{-1}$, $\rho_{-1}^{(i)} = {\rho^{(i)}}^{-1}$, and $s_{-1}^{(i)} = -{\rho^{(i)}}^{-1}{S^{(i)}}^{-1}r^{(i)}$. Therefore

$$\hat{r}_n^{(i)} \equiv \begin{bmatrix} s_{-1}^{(i)} \\ \rho_{-1}^{(i)} \end{bmatrix}.$$

Moreover,

$$({T^{(i)}}^T T^{(i)})^{-1} = R_{-1}^{(i)} {R_{-1}^{(i)}}^T = \begin{bmatrix} S_{-1}^{(i)} {S_{-1}^{(i)}}^T + s_{-1}^{(i)} {s_{-1}^{(i)}}^T & \rho_{-1}^{(i)} s_{-1}^{(i)} \\ \rho_{-1}^{(i)} {s_{-1}^{(i)}}^T & {\rho_{-1}^{(i)}}^2 \end{bmatrix}.$$

Therefore, the last column of $({T^{(i)}}^T T^{(i)})^{-1}$, is

$$({T^{(i)}}^T T^{(i)})^{-1} e_n = R_{-1}^{(i)} {R_{-1}^{(i)}}^T e_n = \rho_{-1}^{(i)} \begin{bmatrix} s_{-1}^{(i)} \\ \rho_{-1}^{(i)} \end{bmatrix}.$$

On the other hand,

$$\begin{aligned} ({T^{(i)}}^T T^{(i)})^{-1} e_n &= U^{(i)} {\Lambda^{(i)}}^{-2} {U^{(i)}}^T e_n \\ &= [u_1^{(i)}, \ldots, u_{n-1}^{(i)}, u_n^{(i)}] \begin{bmatrix} \frac{1}{{\lambda_1^{(i)}}^2} & & & \\ & \ddots & & \\ & & \frac{1}{{\lambda_{n-1}^{(i)}}^2} & \\ & & & \frac{1}{{\lambda_n^{(i)}}^2} \end{bmatrix} [u_1^{(i)}, \ldots, u_{n-1}^{(i)}, u_n^{(i)}]^T e_n \\ &= \frac{{u_1^{(i)}}^T e_n}{{\lambda_1^{(i)}}^2} u_1^{(i)} + \cdots + \frac{{u_{n-1}^{(i)}}^T e_n}{{\lambda_{n-1}^{(i)}}^2} u_{n-1}^{(i)} + \frac{{u_n^{(i)}}^T e_n}{{\lambda_n^{(i)}}^2} u_n^{(i)}. \end{aligned}$$

Thus if $\lambda_n^{(i)} \to 0$,

$$({T^{(i)}}^T T^{(i)})^{-1} e_n = \rho_{-1}^{(i)} \begin{bmatrix} s_{-1}^{(i)} \\ \rho_{-1}^{(i)} \end{bmatrix} \to \frac{{u_n^{(i)}}^T e_n}{{\lambda_n^{(i)}}^2} u_n^{(i)}.$$

□

Thus, the last column of $R_{-1}^{(i)}$ becomes more and more parallel to $u_n^{(i)}$ as $\lambda_n^{(i)}$ goes to 0.

REMARK 4.1. *There is no loss of generality assuming the strict inequality between the two smallest eigenvalues of $T^{(i)}$ in (4.1). In fact, if the smallest k eigenvalues of $T^{(i)}$ are equal, $k \leq n$, and $\lambda_j^{(i)} \to 0, j = n-k+1, n-k+2, \ldots, n-1, n$, then the last k columns of $({T^{(i)}}^T T^{(i)})^{-1}$ approximate the eigenvectors corresponding to $\lambda_j^{(i)}, j = n-k+1, n-k+2, \ldots, n-1, n$, i.e.,*

$$\begin{aligned} ({T^{(i)}}^T T^{(i)})^{-1} [e_{n-k+1}, \ldots, e_{n-1}, e_n] &= R_{-1}^{(i)} {R_{-1}^{(i)}}^T [e_{n-k+1}, \ldots, e_{n-1}, e_n] \\ &\to W_k [u_{n-k+1}^{(i)}, \ldots, u_{n-1}^{(i)}, u_n^{(i)}], \end{aligned}$$

where $W_k \in \mathbb{R}^{k \times k}$ is nonsingular.

¿From a theoretical point of view, the described GSA runs to the completion if the matrix ${T^{(i)}}^T T^{(i)}$ is symmetric positive definite. As $i \to \infty$, the matrix $T^{(i)}$ tends to become singular and $\lambda_n^{(i)} \to 0$.

Suppose $\lambda_{n-1}^{(i)} > \lambda_n^{(i)} \approx 0$. The generalized Schur algorithm for computing simultaneously the R factor of $T^{(i)}$ and its inverse breaks down after the $(n-1)$–th iteration.

In particular, if $\lambda_n^{(i)} = 0$, in the beginning of the nth iteration of GSA, it turns out

$$\|G(n, 1:2)\|_2 = \|G(n, 3:4)\|_2.$$

Therefore, after steps (A.2), (A.3), (A.4) and (A.5) of the n–th iteration of GSA, we have

$$G(n, 1) = G(n, 3).$$

As a consequence, the hyperbolic rotation at step (A.10) of the nth iteration can not be computed, generating a break–down in the algorithm [19, 7].

We will show that, this is a *"lucky"* break–down because the eigenvector corresponding to $\lambda_n^{(i)} = 0$ can be easily retrieved. The following lemma holds.

LEMMA 4.2. *Suppose $\lambda_{n-1}^{(i)} > \lambda_n^{(i)} = \varepsilon_1$, with ε_1 very small. At the n–th step of Algorithm 3, after steps (A.2), (A.3), (A.4) and (A.5) of the n–th iteration of GSA, we have*

$$G(n, 1) = G(n, 3) + \varepsilon_2, \quad \text{with} \quad \varepsilon_2 \sqrt{2G(n, 3)^2 + \varepsilon_2} \le \varepsilon_1. \tag{4.3}$$

Proof. Taking (4.2) into account,

$$\begin{aligned} T^{(i)} {T^{(i)}}^T &= R^{(i)} {R^{(i)}}^T \\ &= \begin{bmatrix} S^{(i)} {S^{(i)}}^T + r^{(i)} {r^{(i)}}^T & \rho^{(i)} r^{(i)} \\ \rho^{(i)} {r^{(i)}}^T & {\rho^{(i)}}^2 \end{bmatrix} \\ &= \begin{bmatrix} S^{(i)} {S^{(i)}}^T & \\ & 0 \end{bmatrix} + \begin{bmatrix} r^{(i)} \\ \rho^{(i)} \end{bmatrix} \begin{bmatrix} {r^{(i)}}^T, & \rho^{(i)} \end{bmatrix}. \end{aligned}$$

Denoted by $Q_S \Lambda_S Q_S^T$ the spectral decomposition of $S^{(i)} {S^{(i)}}^T$ with $Q_S \in \mathbb{R}^{(n-1)\times(n-1)}$ orthogonal, and $\Lambda_S = \operatorname{diag}(\mu_1, \mu_2, \dots, \mu_{n-1})$, $\mu_1 \ge \mu_2 \ge \dots \ge \mu_{n-1} \ge 0$, then

$$T^{(i)} {T^{(i)}}^T = \begin{bmatrix} Q_S & \\ & 1 \end{bmatrix} \left(\begin{bmatrix} \Lambda_S & \\ & 0 \end{bmatrix} + \begin{bmatrix} z^{(i)} \\ \rho^{(i)} \end{bmatrix} \begin{bmatrix} {z^{(i)}}^T, & \rho^{(i)} \end{bmatrix} \right) \begin{bmatrix} Q_S & \\ & 1 \end{bmatrix}^T,$$

with $z^{(i)} = Q_S^T r^{(i)} \equiv \begin{bmatrix} \zeta_1 & \zeta_2 & \dots & \zeta_{n-1} \end{bmatrix}^T$. Thus, the eigenvalues of $T^{(i)} {T^{(i)}}^T$ are the zeros of the secular equation [8],

$$f(\lambda) = 1 + \sum_{j=1}^{n-1} \frac{\zeta_j^2}{\mu_j - \lambda} - \frac{{\rho^{(i)}}^2}{\lambda}.$$

Since

$$f({\lambda_n^{(i)}}^2) = 1 + \sum_{j=1}^{n-1} \frac{\zeta_j^2}{\mu_j - {\lambda_n^{(i)}}^2} - \frac{{\rho^{(i)}}^2}{{\lambda_n^{(i)}}^2} = 0,$$

it turns out

$$\rho^{(i)^2} = \lambda_n^{(i)^2}\left(1+\sum_{j=1}^{n-1}\frac{\zeta_j^2}{\mu_j-\lambda_n^{(i)^2}}\right) \le \lambda_n^{(i)^2}. \tag{4.4}$$

On the other hand, by (4.3),

$$\begin{aligned}\rho^{(i)^2} &= G(n,1)^2 - G(n,3)^2\\ &= \varepsilon_2\left(2G(n,3)^2+\varepsilon_2\right).\end{aligned} \tag{4.5}$$

Hence (4.3) follows from (4.4) and (4.5). □

At step (A.5) of the n–th iteration of GSA, the corresponding hyperbolic rotation to apply to $G(n:2n,1)$ and $G(n:2n,3)$ is

$$H_n = \frac{G(n,3)}{\sqrt{\varepsilon_2^2+2G(n,3)\varepsilon_2}}\begin{bmatrix} 1+\frac{\varepsilon_2}{G(n,3)} & -1\\ -1 & 1+\frac{\varepsilon_2}{G(n,3)}\end{bmatrix}.$$

As $\lambda_n^{(i)} = \varepsilon_1$ becomes closer and closer to 0, and, as a consequence, ε_2 becomes closer and closer to 0, the last column of $R_{-1}^{(i)}$, i.e., the vector made up by the entries of $G(:,1)$ from $n+1$ up to $2n$ after the multiplication of the matrix having $G(:,1)$ and $G(:,3)$ as columns, by H_n, becomes more and more parallel to $u_n^{(i)}$. Since $\tau u_n^{(i)}$ is also an eigenvector of $T^{(i)}$ corresponding to $\lambda_n^{(i)}$, with $\tau \neq 0$, the same approximation of the eigenvector can be obtained if step (5) of the n–th step of GSA, replacing H_n in the multiplication by the first and third generator columns, by the matrix

$$\tilde{H}_n = \begin{bmatrix} 1+\frac{\varepsilon_2}{G(n,3)} & -1\\ -1 & 1+\frac{\varepsilon_2}{G(n,3)}\end{bmatrix}.$$

If $\lambda_n^{(i)} = 0$, the matrix $\tilde{H}_i$ becomes

$$\tilde{H}_n = \begin{bmatrix} 1 & -1\\ -1 & 1\end{bmatrix}, \tag{4.6}$$

i.e., a well–conditioned one. Thus, a multiple of the eigenvector corresponding to $\lambda_n^{(i)} = 0$ can be easily retrieved replacing the computation of the elements of the hyperbolic rotation in step (A.11) of Algorithm 3 (GSA) by the elements of the matrix in (4.6).

4.1. Properties of the eigenvectors of symmetric Toeplitz matrices. Let J_p be the *exchange* matrix of order n, i.e.,

$$J_p = \begin{bmatrix} & & & 1\\ & & 1 & \\ & \iddots & & \\ 1 & & & \end{bmatrix}.$$

A matrix $A \in \mathbb{R}^{n\times n}$ is said *centrosymmetric* if $A = J_pAJ_p$. An *even* vector $v \in \mathbb{R}^n$ is a vector such that

$$J_p v = v.$$

An *odd* vector $w \in \mathbb{R}^n$ is a vector such that

$$J_p w = -w.$$

Given a real symmetric centrosymmetric matrix $A \in \mathbb{R}^{n\times n}$, with $A = U\Lambda U^T$ its eigendecomposition, it is well–known [2] that, $\lfloor \frac{n}{2} \rfloor$ eigenvectors of U are even and $n - \lfloor \frac{n}{2} \rfloor$ eigenvectors of U are odd.

Therefore, if x is a "good" approximation of an eigenvector u of T, the sign structure of its entries can be either odd or even. Thus, the following scheme usually yields a better approximation y of the eigenvector than x.

```
function (x) = odd_even(x, n);
for i = 1 : ⌊n/2⌋,
    t = (|x(i)| + |x(n+1-i)|) /2;
    y(i) = sign(x(i))t;
    y(n+1-i) = sign(x(n+1-i))t;
end
```

4.2. Iterative refinement. At each step of algorithm 4, the R factor and its inverse of the QR factorization of $T^{(i)}$ are computed. The latter algorithm ends yielding an approximation of the smallest eigenvalue of T, and an approximation of the corresponding eigenvector in the last column of $R_{-1}^{(i)}$. Therefore, the obtained approximations can be refined via the following scheme.

```
% Iterative refinement
function (x) = iter_refine(R, x, l);
% INPUT:   R, the R factor of the QR factorization of T
%          x, an approximation of the eigenvector
%             corresponding to the smallest eigenvalue
%          l, number of iterations
% OUTPUT:  x, an approximation of the eigenvector corresponding
%             to the smallest eigenvalue

for i = 1 : l,
    x = R^T\x;
    x = R\x;
    x = x/||x||_2;
end
```

Usually, two iterations of iterative refinement at most are needed to compute the eigenvector corresponding to the smallest eigenvalue of T in an accurate way. The computational complexity of each iteration of iterative refinement is n^2.

In the sequel, the proposed algorithm is described in a matlab–like fashion.

```
% ALGORITHM 5. Computation of the smallest eigenvalue of
%              of a SPD Toeplitz matrix T.
% INPUT:   T, a SPD Toeplitz matrix of order n,
%          tol_1, tol_2, fixed tolerances,
% OUTPUT:  λ_n^(N), the smallest singular value of T;
%          z_1^(o), the corresponding eigenvector.

function [λ_n^(N), z_1^(o)] =small_eig_T(T, n, tol_1, tol_2);
(B.1)   λ_n^(N) = 0;
```

(B.2) $Iflag = 0;$
(B.3) $R_0 = 0;\ z_1^{(o)} = 0;\ z_2^{(o)} = 0;\ z_3^{(o)} = 0;$
(B.4) $iter = 0;$
(B.5) while $Iflag == 0,$
(B.6) $\quad iter = iter + 1;$
(B.7) $\quad [G] = \texttt{gener}(T);$
(B.8) $\quad (R_1, R_1^{-1}, z_1^{(n)}, z_2^{(n)}) = \texttt{schur}(G);$
(B.9) $\quad$ if $R_1(n,n) > \texttt{tol}_1$ & $\lambda_{inc} > \texttt{tol}_2$
(B.10) $\quad\quad R_0 = R_1;\ z_1^{(o)} = z_1^{(n)};\ z_2^{(o)} = z_2^{(n)};$
(B.11) $\quad\quad [\lambda_{inc}] = \texttt{fassino\_adapted}(R_0, R_0^{-1});$
(B.12) $\quad\quad \lambda_n^{(N)} = \lambda_n^{(N)} + \lambda_{inc};$
(B.13) $\quad\quad T = T - \lambda_n^{(N)} I_n;$
(B.14) $\quad$ else
(B.15) $\quad\quad Iflag = 1;$
(B.16) $\quad\quad$ if $iter == 1,$
(B.17) $\quad\quad\quad [\lambda^{(N)}] = \texttt{newton\_smallest}(T, 0);$
(B.18) $\quad\quad$ else
(B.19) $\quad\quad\quad z_1^{(o)} = z_1^{(o)} - z_2^{(o)};$
(B.20) $\quad\quad\quad [z_1^{(o)}] = \texttt{odd\_even}(z_1^{(o)});$
(B.21) $\quad\quad\quad [z_1^{(o)}] = \texttt{iter\_refine}(R_0, z_1^{(o)}, l);$
(B.22) $\quad\quad\quad [z_1^{(o)}] = \texttt{odd\_even}(z_1^{(o)});$
(B.23) $\quad\quad\quad \lambda^{(N)} = ({z_1^{(o)}}^T T z_1^{(o)})/({z_1^{(o)}}^T T z_1^{(o)});$ % Rayleigh quotient
(B.24) $\quad\quad$ end
(B.25) $\quad$ end
(B.26) end

The sequence of the values λ_{inc}, generated in (11), decreases with the number of iterations. The algorithm stops if $R_1(n,n) < \texttt{tol}_1$ or $\lambda_{inc} < \texttt{tol}_2$. In the numerical experiments, $\texttt{tol}_1 = \sqrt{n} \times 10^{-6}$ and $\texttt{tol}_2 = 7.8 \times 10^{-8}$. The algorithm fails to compute in an accurate way the smallest eigenvalue if it is smaller than $\texttt{tol}_2$ and $R_1(n,n) < \texttt{tol}_1$. In this case, the smallest eigenvalue of T and the corresponding eigenvector are computed in (B.17) by the Newton method described in [18, 17], since it requires very few iterations if $\lambda_n \leq 7.8 \times 10^{-8}$. The vectors $z_1^{(n)}$ and $z_2^{(n)}$ are made by the entries $n+1$ up to $2n$ of the generators $G(:,1)$ and $G(:,3)$, respectively, before step (A.11) of the nth iteration of Algorithm 4. In the row (B.19), the approximation of the eigenvector corresponding to λ_n is computed as the first column of the product of $[z_1^{(o)}, z_2^{(o)}]$ by the matrix in (4.6). Once the approximation of the eigenvector is computed ((B.21) and (B.22)), the final approximation of the smallest eigenvalue of T can be obtained as the Rayleigh quotient (B.23).

5. Numerical examples. In this section we present some numerical results. We tested Algorithm 5 with the matrices considered in [5]. More precisely, we generated the following random SPD Toeplitz matrices[‡]

$$T_n = m \sum_{k=1}^{n} w_k T_{2\pi\theta_k}, \tag{5.1}$$

[‡]These matrices can be generated by the matlab gallery toolbox by using toeppd as an input parameter.

where n is the dimension, m is chosen so that T is normalized in order to have the entries on the main diagonal of the matrix equal to 1,

$$T_\theta = (t_{ij})_= (\cos((i-j)\cdot\theta)),$$

and w_k and θ_k are uniformly distributed random numbers taken from $[0,1]$ generated by the `matlab` function `rand`.

For each $n = 128, 256, 512, 1024$, we generated 100 SPD Toeplitz matrices of type (5.1). For each Toeplitz matrix, we construct a matrix $\hat{Q} \in \mathbb{R}^{n\times n}$, having as first $n-1$ columns the eigenvectors corresponding to the largest $n-1$ eigenvalues, computed by the function `eig` of `matlab`, and as last column the eigenvector corresponding to the smallest eigenvalue, computed by Algorithm 5, respectively.

In Table 5.1 we report the average of the absolute errors, the relative errors, the orthogonality of the matrix $\hat{Q}$, and the number of iterations, respectively, computing the approximation $\lambda_n^{(N)}$ by the proposed algorithm, supposing as the exact smallest eigenvalue λ_n, the one computed by the matlab function `eig`. The results show that

n	$\lvert\lambda_n - \lambda_n^{(N)}\rvert$	$\lvert\lambda_n - \lambda_n^{(N)}\rvert/\lambda_n$	$\lVert\hat{Q}^T\hat{Q}^T - I_n\rVert_2/n$	% Iterations
128	3.84×10^{-16}	8.52×10^{-12}	1.24×10^{-11}	4.35
256	3.96×10^{-16}	1.37×10^{-11}	6.90×10^{-11}	4.46
512	8.34×10^{-16}	2.24×10^{-11}	2.31×10^{-10}	4.63
1024	2.19×10^{-15}	5.94×10^{-11}	8.46×10^{-10}	4.74

TABLE 5.1

Average of the absolute errors, the relative errors, orthogonality of the matrix $\hat{Q}$, and the number of iterations, respectively, computing the approximation $\lambda_n^{(N)}$ by the proposed algorithm

the proposed algorithm computes accurately the smallest eigenvalue of SPD Toeplitz matrices.

6. Conclusions. An algorithm for computing the smallest eigenvalue of symmetric positive definite Toeplitz matrices is presented in this paper. It relies on an algorithm proposed in [6] to compute a lower bound of the smallest singular value of full-rank rectangular matrices and a suitable version of the generalized Schur algorithm.

REFERENCES

[1] A.W. Bojańczyk, R.P. Brent, P. Van Dooren, F.R. de Hoog, *A note on downdating the Cholesky factorization,* SIAM J. Sci. Statist. Comput. 8 (1987), no. 3,, 210–221.

[2] A. Cantoni, F. Butler, *Eigenvalues and eigenvectors of symmetric centrosymmetric matrices*, Linear Algebra Appl. 13 (1976), 275-288.

[3] S. Chandrasekaran, A. Sayed, *Stabilizing the Generalized Schur Algorithm,* SIAM J. Matrix Anal. Appl. 17 (1996), no. 4, 950–983.

[4] G. Cybenko, *On the eigenstructure of Toeplitz matrices,* IEEE Trans. Acoust. Speech Signal Proc., ASSP, 31 (1984), pp. 910–920.

[5] G. Cybenko and C. F. Van Loan, *Computing the minimum eigenvalue of a symmetric positive definite Toeplitz matrix,* SIAM J. Sci. Stat. Comp., 7 (1986), pp 123–131.

[6] C. Fassino, *On updating the least singular value: a lower bound,* Calcolo 40 (2003), no. 4, 213–229.

[7] K. Gallivan, A. Vandendorpe, and P. Van Dooren, *Model reduction via truncation: an interpolation point of view*, Linear Algebra Appl., 375 (2003), 115–134.

[8] G. H. Golub, Some Modified Matrix Eigenvalue Problems, SIAM Review 15 (1973) pp. 275-584.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations,* Third ed., The John Hopkins University Press, Baltimore, MD, 1996.
[10] Y.H. Hu and S.Y. Kung, *Computation of the minimum eigenvalue of a Toeplitz matrix by the Levinson algorithm,* Proocedings SPIE 25th International Conference (Real Time Signal Processing), San Diego, August 1980, pp. 40–45.
[11] Y.H. Hu and S.Y. Kung, *A Toeplitz eigensystem solver,* IEEE Trans. Acoust. Speech Signal Proc., ASSP 33 (1986), pp. 485–491.
[12] T. Kailath and A. H. Sayed, *Fast reliable algorithms for matrices with structure*, SIAM, Philadelphia, USA, 1999.
[13] A. Kostić and H. Voss, *Recurrence relations for the even and odd characteristic polynomials of a symmetric Toeplitz matrix,* J. Comput. Appl. Math. 173 (2005), no. 2, 365–369.
[14] A. Kostić and H. Voss, *A method of order* $1+\sqrt{3}$ *for computing the smallest eigenvalue of a symmetric Toeplitz matrix,* WSEAS Trans. Math. 1 (2002), no. 1-4, 1–6.
[15] W. Mackens and H. Voss, *The Minimum Eigenvalue of a Symmetric Positive–Definite Toeplitz Matrix and Rational Hermitian Interpolation,* SIAM J. Matrix Anal. Appl., 18 (1997), pp. 521–534.
[16] W. Mackens and H. Voss, *A projection method for computing the minimum eigenvalue of a symmetric positive definite Toeplitz matrix,* Proceedings of the Sixth Conference of the International Linear Algebra Society (Chemnitz, 1996). Linear Algebra Appl. 275/276 (1998), 401–415.
[17] W. Mackens and H. Voss, *Computing the minimum eigenvalue of a symmetric positive definite Toeplitz matrix by Newton-type methods,* SIAM J. Sci. Comput. 21 (1999/00), no. 4, 1650–1656.
[18] N. Mastronardi and D. Boley, *Computing the smallest eigenpair of a symmetric positive definite Toeplitz matrix,* SIAM J. Sci. Comput. 20 (1999), no. 5, 1921–1927.
[19] N. Mastronardi, D. Kressner, V. Sima, P. Van Dooren, and S. Van Huffel, A fast algorithm for state-space identification via exploitation of the displacement structure, J. Comp. Appl. Math., 132(1):71–81, 2001.
[20] N. Mastronardi, M. Van Barel, and R. Vandebril, *A Schur-based algorithm for computing the smallest eigenvalue of a symmetric positive definite Toeplitz matrix*, K.U.Leuven, Department of Computer Science, Report TW 461, May, 2006, accepted for publication in Linear Algebra Appl.
[21] A. Melman, *Computation of the Newton step for the even and odd characteristic polynomials of a symmetric positive definite Toeplitz matrix*, Math. Comp. 75 (2006), no. 254, 817–832.
[22] A. Melman, *Computation of the smallest even and odd eigenvalues of a symmetric positive-definite Toeplitz matrix*, SIAM J. Matrix Anal. Appl. 25 (2004), no. 4, 947–963.
[23] A. Melman, *Extreme eigenvalues of real symmetric Toeplitz matrices*, Math. Comp. 70 (2001), no. 234, 649–669.
[24] A. Melman, *Spectral functions for real symmetric Toeplitz matrices*, J. Comput. Appl. Math. 98 (1998), no. 2, 233–243.
[25] M.K. Ng, *Preconditioned Lanczos methods for the minimum eigenvalue of a symmetric positive definite Toeplitz matrix,* SIAM J. Sci. Comput. 21 (2000), no. 6, 1973–1986.
[26] V.F. Pisarenko, *The retrieval of harmonics from a covariance function,* Geophysical J. Royal Astronomical Soc., 33 (1973), pp. 347–366.
[27] M. Stewart and P. Van Dooren, *Stability issues in the factorization of structured matrices*, SIAM J. Matrix Anal. Appl., 18(1) (1997), pp. 104–118.
[28] W.F. Trench, *Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices,* SIAM J. Matrix Anal. Appl., 10, No. 2 (1989), pp. 135–146.
[29] W.F. Trench, *Numerical solution of the eigenvalue problem for efficiently structured Hermitian matrices,* Linear Algebra Appl., 154–156 (1991), pp. 415–432.
[30] H. Voss, *A symmetry exploiting Lanczos method for symmetric Toeplitz matrices,* Mathematical journey through analysis, matrix theory and scientific computation (Kent, OH, 1999). Numer. Algorithms 25 (2000), no. 1-4, 377–385.
[31] H. Voss, *Symmetric schemes for computing the minimum eigenvalue of a symmetric Toeplitz matrix,* Special issue celebrating the 60th birthday of Ludwig Elsner. Linear Algebra Appl. 287 (1999), no. 1-3, 359–371.