A Schur–based algorithm for computing the smallest eigenvalue of a symmetric positive definite Toeplitz matrix

Nicola Mastronardi   *Marc Van Barel*
Raf Vandebril

Report TW 461, May 2006

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

A Schur–based algorithm for computing the smallest eigenvalue of a symmetric positive definite Toeplitz matrix

Nicola Mastronardi   *Marc Van Barel*
Raf Vandebril

Report TW 461, May 2006

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 symmetric positive definite Toeplitz matrices. 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 the smallest eigenvalue of a symmetric positive definite Toeplitz matrix is derived. The algorithm relies on the computation of the R factor of the QR–factorization of the Toeplitz matrix and the inverse of R. The latter computation is efficiently accomplished by the generalized Schur algorithm.

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 SCHUR–BASED 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 symmetric positive definite Toeplitz matrices. 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 the smallest eigenvalue of a symmetric positive definite Toeplitz matrix is derived. The algorithm relies on the computation of the R factor of the QR–factorization of the Toeplitz matrix and the inverse of R. The latter computation is efficiently accomplished by the generalized Schur algorithm.

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.

Recent progress in signal processing and estimation has generated considerable interest in the problem of computing the smallest eigenvalue of a symmetric positive definite (SPD) Toeplitz matrix T.

Given the covariance sequence of observed data, Pisarenko [23] suggested a method to determine the sinusoidal frequencies from the eigenvector associated to the smallest eigenvalue of the covariance matrix.

Many algorithms have been proposed in the literature to compute the smallest eigenvalue of T [12, 13, 4, 5, 18, 16, 28, 19, 22, 25, 26].

The algorithm proposed in [5] is based on using bisection to find a point between the smallest eigenvalue of the $n \times n$ SPD Toeplitz matrix T and the smallest one of its $(n-1) \times (n-1)$ leading principal submatrix, and then to improve that point using Newton's method for the determination of a zero of the secular equation. Newton's method was replaced by more appropriate root finders based on Hermitian rational interpolation in the algorithms presented in [18, 16].

Newton's method applied to the characteristic polynomial of T was proposed in [21] and related algorithms were proposed in [27, 20]. All the latter algorithms rely

*The research of the first author was partially supported by MIUR, grant number 2004015437, and by the National Research Council of Italy under the Short Term Mobility Program. 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), Center of Excellence: Optimization in Engineering, 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 Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, project IUAP V-22 (Dynamical Systems and Control: Computation, Identification & Modelling). The scientific responsibility rests with the authors.

†Istituto per le Applicazioni del Calcolo "M. Picone", sez. 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, Celestijnenlaan200A, B-3001 Leuven (Heverlee), Belgium. email: {Raf.Vandebril, Marc.VanBarel, Nicola.Mastronardi}@cs.kuleuven.be

on the Levinson–Durbin solutions of sequences of Yule–Walker systems.

In all the proposed methods, the most expensive part is the computation of a region in which the proposed algorithms monotonically converge to the desired eigenvalue.

An algorithm to compute the smallest singular value and the corresponding left and right singular vectors of a Toeplitz matrix was proposed in [15].

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 the lower bound of the smallest singular value $\hat{\sigma}$ of the augmented matrix $[A|b]$ is proposed in [9]. Quite often, this algorithm yields a reliable lower bound 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 rectangular matrix, based on the QR–factorization, is proposed in the latter paper.

An upper bound for the smallest singular value of matrices can be computed by an algorithm described in [8]. Combining it with the algorithm in [9], a narrow interval in which the smallest singular value of matrices lies can be computed.

In this paper, taking these recent results into account, we propose a fast algorithm that determines a narrow interval in which the smallest eigenvalue of a SPD Toeplitz matrix lies. In fact, we show that the algorithms described in [9, 8] can be efficiently combined with a suitable variant of the generalized Schur algorithm (GSA) [14] to estimate a lower and an upper bound of the smallest eigenvalue of SPD Toeplitz matrices.

The paper is organized as follows. In § 2 the algorithm described in [9] for computing the lower bound of the smallest singular value of the augmented matrix $[A|b]$ is shortly introduced, followed by § 3, describing the algorithm in [8], yielding an upper bound of the smallest singular value of matrices. The particular version of GSA is described in § 4. The proposed algorithm is described in § 5. The numerical examples are reported in § 6, followed 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 [9], known a lower bound of the smallest singular value $\sigma(A)$ of A. It relies on the following

THEOREM 2.1. *[9] 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.

%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[δ] =fassino(R);

δ = |R(1,1)|;
for k = 1 : n − 1,
    X(1 : k, k + 1) = R^{-1}(1 : k, 1 : k)R(1 : k, k + 1);
    ρ = |R(k + 1, k + 1)|;
    δ = δ sqrt(E(δ, X(1 : k, k + 1), ρ));
end
```

The algorithm requires to compute the R factor of the QR–factorization of A. This is accomplished with standard techniques, e.g., see [11], 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. *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*

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

Proof. Since R is nonsingular upper triangular, the inverse of the upper–left leading principal submatrix $R(1:k,1:k)$ is equal to the upper–left leading principal submatrix of order k of the inverse. Let $e_j^{(n)}$ be the j–th vector of the canonical basis of $\mathbb{R}^n$, $j=1,2,\ldots,n$.

Since $e_{k+1}^{(n)} = R^{-1}Re_{k+1}^{(n)}$, then

$$\begin{cases} R^{-1}(1:k,1:k)R(1:k,k+1) + R(k+1,k+1)R^{-1}(1:k,k+1) = [\underbrace{0,\ldots,0}_{k}]^T \\ R^{-1}(k+1,k+1)R(k+1,k+1) = 1, \end{cases}$$

from which (2.1) follows. □

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^{-1}, with R the R factor of the QR–factorization of A.
%Output: δ, a lower bound of the smallest singular value of A.

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

```
function[δ] =fassino_adapted(R^{-1});

δ = |R(1,1)|;
for k = 1 : n − 1,
    X(1 : k, k + 1) = −R^{-1}(k + 1, k + 1)R^{-1}(1 : k, k + 1);
    ρ = |R(k + 1, k + 1)|;
    δ = δ sqrt(E(δ, X(1 : k, k + 1), ρ));
end
```

In Section 4 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. Computing an upper bound for the smallest singular value. Some algorithms for computing an upper bound for the smallest singular value of a triangular matrix R are described in [1, 8]. In this section we shortly describe an algorithm for computing a lower bound of the largest singular value, i.e., the spectral norm, of triangular matrices by approximating the right singular vector, corresponding to this largest singular value described in [1]. An upper bound of the smallest singular value is obtained applying the algorithm to the inverse of the upper triangular matrix.

Combining this algorithm with Algorithm 1 or 2, an algorithm for computing a tight interval in which the smallest singular value lies can be derived.

Let $R \in \mathbb{R}^{n\times n}$ be an upper triangular matrix and let $\tilde{z}$ be an approximation of unit length of the right singular vector $\overline{z}$ corresponding to the largest singular value of R,

$$\overline{z} = \arg \max_{\|z\|_2=1} \|Rz\|_2. \tag{3.1}$$

Let

$$\hat{R} = \begin{bmatrix} R & v \\ & \gamma \end{bmatrix},$$

be the augmented upper triangular matrix obtained from R adding the column $[v^T, \gamma]^T \in \mathbb{R}^{n+1}$. The aim is to find an approximation $\hat{z}$ of the right singular vector of $\hat{R}$ corresponding to the largest singular value of the form

$$\hat{z} = \begin{bmatrix} s\tilde{z} \\ c \end{bmatrix}, \quad \text{with} \quad c^2 + s^2 = 1. \tag{3.2}$$

Since

$$\|\hat{R}\hat{z}\|_2^2 = \begin{bmatrix} s & c \end{bmatrix} \begin{bmatrix} \tilde{z}^T R^T R\tilde{z} & \tilde{z}^T R^T v \\ v^T R\tilde{z} & v^T v + \gamma^2 \end{bmatrix} \begin{bmatrix} s \\ c \end{bmatrix},$$

the solution $\hat{z}$ of (3.1) with constraint (3.2) can be computed analytically,

$$\hat{z} = \begin{bmatrix} \hat{s}\tilde{z} \\ \hat{c} \end{bmatrix}, \quad \text{with} \quad \begin{bmatrix} \hat{s} \\ \hat{c} \end{bmatrix} = \frac{u}{\|u\|_2}, \quad u = \begin{bmatrix} \epsilon^2 - \tau^2 + \sqrt{(\epsilon^2 - \tau^2)^2 + 4\beta^2} \\ 2\beta \end{bmatrix},$$

$$\beta = v^T R\tilde{z}, \quad \epsilon = \|R\tilde{z}\|_2, \quad \tau^2 = v^T v + \gamma^2,$$

and

$$\hat{\epsilon} = \|\hat{R}\hat{z}\|_2 = \sqrt{\hat{s}^2\epsilon^2 + 2\hat{s}\hat{c}\beta + \hat{c}^2\tau^2}.$$

We observe that the matrix–vector product computation can be avoided, exploiting the following recurrence relation,

$$\hat{R}\hat{z} = \begin{bmatrix} \hat{s}R\tilde{z} + \hat{c}v \\ \hat{c}\gamma \end{bmatrix}.$$

Based on the latter considerations, an algorithm for estimating a lower bound of the largest singular value of an upper triangular matrix can be easily derived.

% Algorithm 3. Lower bound of the largest singular value
% of an upper triangular matrix R.
% Input: R, an upper triangular matrix of order n.
% Output: ϵ, a lower bound of the largest singular value of R.
`function`$[\epsilon]$ $=$`duff`(R);

$\epsilon = |R(1,1)|$;
$Rz = R(1,1)$;
for $k = 1 : n-1$,
  $v = R(1:k, k+1)$;
  $\beta = v^T Rz$;
  $\gamma = R(k+1, k+1)$;
  $\tau^2 = v^T v + \gamma^2$;
  $u = \left[\ \epsilon^2 - \tau^2 + \sqrt{(\epsilon^2-\tau^2)^2 + 4\beta^2},\ 2\beta\ \right]^T$
  $u = u/\|u\|_2$;
  $s = u(1)$;
  $c = u(2)$;
  $\epsilon = \sqrt{s^2\epsilon^2 + 2sc\beta + c^2\tau^2}$;
  update of Rz;
end

We observe that Rz is a vector. At step k, of Algorithm 3, the inner products $v^T Rz$ and $v^T v$ must be computed with $2k-1$ floating point operations, respectively. Updating of the vector Rz requires $3k$ operations. Therefore, the computational complexity of Algorithm 3 is $4n^2$.

4. Generalized Schur Algorithm. Let

$$T = \begin{bmatrix} t_1 & t_2 & \ddots & t_n \\ t_2 & t_1 & \ddots & \ddots \\ \ddots & \ddots & \ddots & t_2 \\ t_n & \ddots & t_2 & t_1 \end{bmatrix}. \tag{4.1}$$

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, 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

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

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., [14].

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 $\Phi = Z \oplus Z$. 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 the *generator* matrix.

Denote $v = T^T(T(:,1))$. The columns of G are given by

$$G(:,1) = \frac{1}{\sqrt{v(1)}} \left[\begin{array}{c} v \\ \hline e_1^{(n)} \end{array}\right], \; G(:,2) = \left[\begin{array}{c} 0 \\ t_2 \\ \vdots \\ t_n \\ \hline 0 \\ \vdots \\ 0 \end{array}\right], \; G(:,3) = \left[\begin{array}{c} 0 \\ G(2:n,1) \\ \hline 0 \\ \vdots \\ 0 \end{array}\right], \; G(:,4) = \left[\begin{array}{c} 0 \\ t_n \\ \vdots \\ t_2 \\ \hline 0 \\ \vdots \\ 0 \end{array}\right].$$

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:

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

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,

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

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

[†]Hyperbolic rotations can be computed in different ways. For "stable" implementations see [2, 3].

of the generator matrix G is $4 \ll n$, the GSA for computing R and R^{-1} has $O(n^2)$ computational complexity and it can be summarized in the following algorithm.

```
% ALGORITHM 4. 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}] =schur(G);

for k = 1 : n,
    [c_G, s_G] = giv(G(k,1), G(k,2));
    G(k : n+k, 1 : 2) = G(k : n+k, 1 : 2) [c_G s_G; -s_G c_G];
    [c_G, s_G] = giv(G(k,3), G(k,4));
    G(k : n+k-1, 3 : 4) = G(k : n+k-1, 3 : 4) [c_G s_G; -s_G c_G];
    [c_H, s_H] = hyp(G(k,1), G(k,3));
    G(k : n+k, [1,3]) = G(k : n+k, [1,3]) [c_H -s_H; -s_H c_H];
    R(k, k : n) = G^T(k : n, 1);
    R^{-1}(1 : k, k) = G(n+1 : n+k, 1);
    G(:,1) = ΦG(:,1);
end
```

Each iteration of the latter algorithm involves two products of Givens rotations by a $n \times 2$ matrix, each of those can be accomplished with $6n$ floating point operations, followed by the product of an hyperbolic rotation by a $n \times 2$ 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.

5. Computation of the smallest eigenvalue of a symmetric positive definite Toeplitz matrix. Known the inverse R^{-1} of the R factor of the QR–factorization of a full–rank matrix A, Algorithm 2, applied to R^{-1}, computes a lower bound for the smallest singular value of R. Algorithm 3, applied to R^{-1} computes a lower bound for the largest singular value of R^{-1}. Hence, we have also an upper bound for the smallest singular value of R. Since the eigenvalues and the singular values of a SPD matrix coincide, the latter procedure yields also an upper bound of the smallest eigenvalue of a SPD matrix. In Section 4 we have shown that the R factor of the QR–factorization and its inverse R^{-1} for a Toeplitz matrix can be efficiently computed by means of GSA. Combining these algorithms leads to the following algorithm for computing tight upper and lower bounds for the smallest eigenvalue of a SPD Toeplitz matrix. Note that the output R of `schur` is not used in the remainder of the algorithm and could be skipped.

```
% ALGORITHM 5. Computation of the smallest eigenvalue of a SPD Toeplitz matrix
%              T.
% INPUT:  T, a SPD Toeplitz matrix of order n,
%         tol, a fixed tolerance,
% OUTPUT:  λ_n^{(-)}, a lower bound of the smallest singular value of T;
%          λ_n^{(+)}, an upper bound of the smallest singular value of T.
```

```
function [λ_n^(-), λ_n^(+)] = small_eig_T(T, tol);
λ_n^(-) = 0;
Iflag = 0;
while Iflag == 0,
    [G] = gener(T);
    (R, R^-1) = schur(G);
    if R(n,n) > tol,
        [λ] = fassino_adapted(R^-1);
        λ_n^(-) = λ_n^(-) + λ;
        T = T - λ_n^(-) I_n;
    else
        [ε] = duff(R^-1);
        λ_n^(+) = λ_n^(-) + 1/ε;
        Iflag = 1;
    end
end
```

Unfortunately, since Algorithms 1 and 2 suffers from the loss of accuracy (see the error analysis in [9]) and because of the weak stability of GSA [24], the threshold tol must be chosen not to small. For the numerical experiments we have chosen tol $= \sqrt{n} \times 10^{-4}$. Let R be an $n \times n$ nonsingular upper triangular matrix. The following theorem shows the relationship between the smallest singular value of the nonsingular augmented upper triangular matrix $\hat{R}$ and the entry of this matrix in position $(n+1, n+1)$.

THEOREM 5.1. *Let R be an $n \times n$ nonsingular upper triangular matrix and $\hat{R}$ a nonsingular augmented upper triangular matrix based on R, i.e.,*

$$\hat{R} = \begin{bmatrix} R & v \\ & \gamma \end{bmatrix} \quad \text{and} \quad \hat{R}^{-1} = \begin{bmatrix} R^{-1} & u \\ & 1/\gamma \end{bmatrix},$$

with

$$u = -\frac{1}{\gamma} R^{-1} v. \tag{5.1}$$

Let $R = U \Sigma V^T$ and $\hat{R} = \hat{U}\hat{\Sigma}\hat{V}^T$ be the singular value decompositions of R and $\hat{R}$, respectively, with $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$, $\hat{\Sigma} = \mathrm{diag}(\hat{\sigma}_1, \hat{\sigma}_2, \cdots, \hat{\sigma}_{n+1})$, $\hat{\sigma}_1 \geq \sigma_1 \geq \hat{\sigma}_2 \geq \sigma_2 \geq \cdots \geq \hat{\sigma}_n \geq \sigma_n \geq \hat{\sigma}_{n+1} > 0$.

Then

$$\hat{\sigma}_{n+1}^2 < \frac{\gamma^2}{1 + \frac{\gamma^2 \|u\|_2^2}{\kappa_2^2(R)}}, \tag{5.2}$$

where $\kappa_2(R)$ is the condition number of R in spectral norm.

Proof. Since

$$\hat{R}\hat{R}^T = \begin{bmatrix} U & \\ & 1 \end{bmatrix} \begin{bmatrix} \Sigma^2 + \tilde{v}\tilde{v}^T & \tilde{v} \\ \tilde{v}^T & \gamma^2 \end{bmatrix} \begin{bmatrix} U^T & \\ & 1 \end{bmatrix},$$

with $\tilde{v} = U^T v$, the eigenvalues of $\hat{R}\hat{R}^T$, i.e., $\hat{\sigma}_i^2$, $i = 1, \ldots, n+1$, are the zeros of the *secular equation* [10, 7],

$$f(x) \equiv 1 + \sum_{i=1}^{n} \frac{\tilde{v}_i^2}{\sigma_i^2 - x} - \frac{\gamma^2}{x}.$$

Since $f(\hat{\sigma}_{n+1}^2) = 0$,

$$\begin{aligned} \frac{\gamma^2}{\hat{\sigma}_{n+1}^2} &= 1 + \sum_{i=1}^{n} \frac{\tilde{v}_i^2}{\sigma_i^2 - \hat{\sigma}_{n+1}^2} \\ &> 1 + \sum_{i=1}^{n} \frac{\tilde{v}_i^2}{\sigma_i^2} \\ &= 1 + \|\Sigma^{-1}\tilde{v}\|_2^2 \\ &\geq 1 + \frac{\|\tilde{v}\|_2^2}{\sigma_1^2} \\ &= 1 + \frac{\|v\|_2^2}{\sigma_1^2} \end{aligned}$$

Moreover, from (5.1), it turns out

$$\|u\|_2 \leq \frac{1}{|\gamma|}\|R^{-1}\|_2\|v\|_2 = \frac{1}{|\gamma|\ \sigma_n}\|v\|_2.$$

Hence,

$$\begin{aligned} \hat{\sigma}_{n+1}^2 &< \frac{\gamma^2}{1 + \frac{\|v\|_2^2}{\sigma_1^2}} \\ &\leq \frac{\gamma^2}{1 + \frac{\gamma^2\|u\|_2^2}{\kappa_2^2(R)}}. \end{aligned}$$

□

As a consequence, the smallest singular value of the augmented matrix is at least smaller than the absolute value of the entry in position $(n+1, n+1)$ of the augmented upper triangular matrix $\hat{R}$.

The speed of convergence of Algorithm 5 to the smallest eigenvalue of SPD Toeplitz matrices is not easy to determine. However, based on the numerical experiments we have seen that the algorithm exhibits a quadratic convergence (see § 6).

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

$$T_n = m \sum_{k=1}^{n} w_k T_{2\pi\theta_k}, \tag{6.1}$$

where n is the dimension, m is chosen so that T is normalized in order to have the entries in the main diagonal of the matrices 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`.

Example 1. In the first example the convergence behavior of the proposed algorithm is emphasized. For each $n = 64, 128, 256, 512$, we generated a SPD Toeplitz matrix of

	$\lambda_{64}(T_{64})$	$\lambda_{128}(T_{128})$
	4.528316520827970e-03	1.105877278442042e-02
Iter.	$\lambda_{64}(T_{64}) - \lambda_{64}^{(-)}(T_{64})$	$\lambda_{128}(T_{128}) - \lambda_{128}^{(-)}(T_{128})$
1	6.224898987010354e-04	5.122631876943164e-03
2	6.578738788397785e-06	1.838635898897332e-03
3	1.317166211170506e-10	3.737925390931134e-04
4		1.308234916439117e-05
5		9.658067384038516e-10
	$\lambda_{256}(T_{256})$	$\lambda_{512}(T_{512})$
	7.340798858234904e-05	1.450156643489054e-04
Iter.	$\lambda_{256}(T_{256}) - \lambda_{256}^{(-)}(T_{256})$	$\lambda_{512}(T_{512}) - \lambda_{512}^{(-)}(T_{512})$
1	1.781210151117608e-05	4.208425291080006e-06
2	1.797324164695707e-06	3.878533654183905e-11
3	5.807584469062700e-09	

TABLE 6.1
Convergence behaviour of the proposed algorithm

kind (6.1). In Table 6.1 we report the difference between the smallest eigenvalue of the Toeplitz matrix, computed by the matlab function `eig` and the approximations of the latter eigenvalue yielded by the proposed algorithm at each iteration. We have repeated this experiment several times for different matrices of kind (6.1), obtaining similar results. So, the convergence behaviour of the proposed algorithm seems to be quadratic.

Example 2. In this second example, the behaviour of the upper bound of the smallest singular value, computed by the proposed algorithm, is analysed. In particular, for each $n = 64, 128, 256, 512$, we consider 100 SPD Toeplitz matrices of kind (6.1). In Table 6.2 we report the number of times the computed upper bound is in the interval $[\lambda_n(T_n), \lambda_{n-1}(T_{n-1})]$ (in percentage). In the first column the order of the SPD Toeplitz matrices is reported. In the second column, the average number of iterations required by the proposed algorithm is reported. As n increases the latter interval becomes narrower and narrower. Nevertheless, the percentage of successes is always quite high. If the computed upper bound belongs to the interval $[\lambda_n(T_n), \lambda_{n-1}(T_{n-1})]$,

n	It. Fassino	% $\lambda_n^{(+)} < \lambda_{n-1}(T_{n-1})$
64	3.40	98 %
128	3.32	97 %
256	3.07	81 %
512	2.64	77 %

TABLE 6.2
Comparison between the computed upper bound of $\lambda_n^{(+)}$ and $\lambda_{n-1}(T_{n-1})$

it can be chosen as starting point for the methods proposed in [5, 20]. If the computed upper bound is bigger than $\lambda_{n-1}(T_{n-1})$, the new approximation of the smallest eigenvalue could be chosen as $(\lambda_n^{(-)} + \lambda_n^{(+)})/2$ and proceeding using a bisection scheme (for more details, see [5]).

Example 3. In this last example, the computed lower bound by the proposed algorithm is chosen as starting guess for the Newton's method applied to the characteristic polynomial as described in [21]. Also for this example, for each $n = 64, 128, 256, 512$, we consider 100 SPD Toeplitz matrices of kind (6.1). As before, in the first column the order of the SPD Toeplitz matrices is reported. In the second column, the average number of iterations required by the proposed algorithm is reported. In column 3, the average number of steps of the Newton's method with initial guess $\lambda_n^{(-)}$, is stated. The Newton's method is applied until the absolute error $\lambda_n(T_n) - \lambda_n^{(-)} < 1.0e-8$ (see [21] for more details). In the 4th column, the average number of steps of the Newton's method with 0 as initial guess, is reported. Also in this case, Newton's method is applied until the absolute error is below $1.0e-8$.

n	It. Fassino	number of steps: $\lambda_n^{(-)}$	number of steps: 0
64	3.40	1.56	8.90
128	3.32	2.24	9.96
256	3.07	2.55	10.02
512	2.64	3.12	10.95

TABLE 6.3

Comparison between the number of Newton-iterations with starting value $\lambda_n^{(-)}$ *compared to starting value* 0

7. 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 [9] to compute a lower bound of the smallest singular value of full-rank rectangular matrices and a suitable version of the generalized Schur algorithm. From the numerical experiments, it can be guessed that the proposed algorithm has a quadratic speed of convergence.

A modified version of the proposed algorithm can be efficiently used to compute the condition number of structured matrices, i.e. tridiagonal, semiseparable matrices diagonal plus semiseparable matrices, rank–structured matrices [6], that is, all kind of matrices whose inverse can be computed in a fast way.

REFERENCES

[1] C.H. Bischof, *Incremental condition estimation,* SIAM J. Matrix Anal. Appl. 11 (1990), no. 2, 312–322.

[2] 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.

[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] S. Delvaux and M. Van Barel, *Structures preserved by matrix inversion*, SIAM J. Matrix Anal. Appl., 2005, to appear.

[7] J.J. Dongarra and D.C. Sorensen, *A fully parallel algorithm for the symmetric eigenvalue problem,* SIAM J. Sci. Stat. Comput., 8 (1987), no. 2, pp. 139–154.

[8] I.S. Duff, C. Vömel, *Incremental norm estimation for dense and sparse matrices*, BIT 42 (2002), no. 2, 300–322.

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

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

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations,* Third ed., The John Hopkins University Press, Baltimore, MD, 1996.

[12] 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.

[13] Y.H. Hu and S.Y. Kung, *A Toeplitz eigensystem solver,* IEEE Trans. Acoust. Speech Signal Proc., ASSP 33 (1986), pp. 485–491.

[14] T. Kailath and A. H. Sayed, *Fast reliable algorithms for matrices with structure*, SIAM, Philadelphia, USA, 1999.

[15] J. Kamm and J.G. Nagy, *A Total Least Squares Method for Toeplitz Systems of Equations*, BIT, 38 (1998), pp. 560–582.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] V.F. Pisarenko, *The retrieval of harmonics from a covariance function,* Geophysical J. Royal Astronomical Soc., 33 (1973), pp. 347–366.

[24] 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.

[25] 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.

[26] W.F. Trench, *Numerical solution of the eigenvalue problem for efficiently structured Hermitian matrices,* Linear Algebra Appl., 154–156 (1991), pp. 415–432.

[27] 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.

[28] 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.