

On computing the eigenvectors of a class of structured matrices

Nicola Mastronardi *Marc Van Barel*
Ellen Van Camp *Raf Vandebril*

Report TW 407, September 2004



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

On computing the eigenvectors of a class of structured matrices

Nicola Mastronardi *Marc Van Barel*
Ellen Van Camp *Raf Vandebril*

Report TW 407, September 2004

Department of Computer Science, K.U.Leuven

Abstract

A real symmetric matrix of order n has a full set of orthogonal eigenvectors. The most used approach to compute the spectrum of such matrices reduces first the dense symmetric matrix into a symmetric structured one, i.e., tridiagonal matrices or semiseparable matrices. This step is accomplished in $O(n^3)$ operations. Once the latter symmetric structured matrix is available, its spectrum is computed in an iterative fashion by means of the QR method in $O(n^2)$ operations.

In principle, the whole set of eigenvectors of the latter structured matrix can be computed by means of inverse iteration in $O(n^2)$ operations.

The blemish in this approach is that the computed eigenvectors may not be numerically orthogonal if clusters are present in the spectrum. To enforce orthogonality the Gram-Schmidt procedure is used, requiring $O(n^3)$ operations in the worst case.

In this paper a fast and stable method to compute the eigenvectors of tridiagonal and semiseparable matrices which does not suffer from the loss of orthogonality due to the presence of clusters in the spectrum is presented. The algorithm requires $O(n^2)$ floating point operations if clusters of small size are present in the spectrum.

Keywords : symmetric matrix, tridiagonal matrix, semiseparable matrix, eigenvalues, eigenvectors.

AMS(MOS) Classification : Primary : 65F15, Secondary : 15A18.

On computing the eigenvectors of a class of structured matrices

N. Mastronardi ^{a,*}, M. Van Barel ^{b,2}, E. Van Camp ^{b,2},
R. Vandebril ^{b,2}

^a*Istituto per le Applicazioni del Calcolo, CNR, via Amendola 122/D, 70126, Bari, Italy*

^b*Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium*

Abstract

A real symmetric matrix of order n has a full set of orthogonal eigenvectors. The most used approach to compute the spectrum of such matrices reduces first the dense symmetric matrix into a symmetric structured one, i.e., tridiagonal matrices or semiseparable matrices. This step is accomplished in $O(n^3)$ operations. Once the latter symmetric structured matrix is available, its spectrum is computed in an iterative fashion by means of the QR method in $O(n^2)$ operations.

In principle, the whole set of eigenvectors of the latter structured matrix can be computed by means of inverse iteration in $O(n^2)$ operations.

The blemish in this approach is that the computed eigenvectors may not be numerically orthogonal if clusters are present in the spectrum. To enforce orthogonality the Gram-Schmidt procedure is used, requiring $O(n^3)$ operations in the worst case.

In this paper a fast and stable method to compute the eigenvectors of tridiagonal and semiseparable matrices which does not suffer from the loss of orthogonality due to the presence of clusters in the spectrum is presented. The algorithm requires $O(n^2)$ floating point operations if clusters of small size are present in the spectrum.

Key words: symmetric matrix, tridiagonal matrix, semiseparable matrix, eigenvalues, eigenvectors.

* *Email address corresponding author:* `n.mastronardi@area.ba.cnr.it`

¹ This work was partially supported by MIUR, grant number 2002014121.

² This research was partially supported by the Research Council K.U.Leuven, project OT/00/16 (SLAP: Structured Linear Algebra Package), by the Fund for Scientific Research–Flanders (Belgium), projects G.0078.01 (SMA: Structured Matrices and their Applications), G.0176.02 (ANCILA: Asymptotic aNalysis of the Con-

1 Introduction

A real symmetric matrix of order n has a full set of orthogonal eigenvectors. The most used approach to compute the spectrum of such matrices reduces first the dense symmetric matrix into a symmetric tridiagonal one [8,9]. This step is accomplished in $O(n^3)$ operations. Once the latter symmetric tridiagonal matrix is available, its spectrum is computed in an iterative fashion by means of the QR method in $O(n^2)$ operations [8,9]. A similar approach has recently been introduced [10,11]. In the latter approach, the role of tridiagonal matrices is played by semiseparable ones, resulting in an algorithm with the same computational complexity [10]. The latter class of matrices shares with the class of symmetric tridiagonal matrices the property that one step of the QR method implicitly shifted can be accomplished in $O(n)$ operations [11] by applying $O(n)$ Givens transformations³.

In principle, the whole set of eigenvectors of the latter matrices can be computed by means of inverse iteration in $O(n^2)$ operations. The blemish in this approach is that the computed eigenvectors may not be numerically orthogonal if clusters are present in the spectrum. To enforce orthogonality the Gram–Schmidt procedure is used, requiring $O(n^3)$ operations in the worst case [3,12].

In this paper, we describe a numerical method to compute the set of the eigenvectors numerically orthogonal. The method requires $O(n^2)$ floating point operations if few clusters of small size are present in the spectrum. The method is based on the fact that the orthogonal matrices generated by one step of the QR method applied to symmetric tridiagonal matrices and symmetric semiseparable ones, are highly structured. For instance, if the matrix is symmetric tridiagonal, the orthogonal matrix generated by one step of the QR method is a upper Hessenberg semiseparable one. Therefore, the product of the latter matrices by a vector can be accomplished in $O(n)$ floating point operations, where n is the size of the involved matrix.

Recently, an efficient $O(n^2)$ algorithm to compute the eigendecomposition of

vergence behavior of Iterative methods in numerical Linear Algebra), G.0184.02 (CORFU: Constructive study of Orthogonal Functions) and G.0455.0 (RHPH: Riemann-Hilbert problems, random matrices and Padé-Hermite approximation), 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.

³ One step of the implicitly shifted QR method is implemented with $n - 1$ Givens rotations if the matrix is symmetric tridiagonal, with $2(n - 1)$ Givens rotations if the matrix is symmetric semiseparable.

symmetric tridiagonal matrices, based on the $Q - -D$ algorithm, has been introduced in [4–7]. A different approach for the same problem has been described in [1], based on the Rayleigh quotient iteration combined with the inverse iteration, with an efficient choice of the initial vectors.

The paper is organized as follows. In section 2 the implicitly shifted QR method is shortly described together with the way to compute the eigenvectors. In section 3 a fast procedure to compute and refine an eigenvector corresponding to a simple eigenvalue is described. The treatment of numerically multiple eigenvalues and the corresponding eigenvectors is handled in section 4 followed by some numerical examples and the conclusions.

2 QR method for symmetric tridiagonal and semiseparable matrices

The implicitly shifted QR method is a powerful tool for computing the eigenvalues of matrices. It can be efficiently implemented for the class of symmetric semiseparable matrices and the class of symmetric tridiagonal ones, and each iteration can be accomplished in $O(n)$ operations by means of $2(n - 1)$ and $n - 1$ Givens similarity transformations, respectively [8,9,11]. The nice feature of the QR method applied to the latter classes of matrices is that the generated matrices belong to the same classes of matrices.

For the sake of simplicity, in this paper we consider only the implicitly shifted QR method applied to symmetric tridiagonal matrices. Similar arguments can be considered for matrices belonging to the class of semiseparable ones.

Let

$$T \equiv T_n^{(0)} = \begin{pmatrix} \alpha_1^{(0)} & \beta_1^{(0)} & & & \\ \beta_1^{(0)} & \alpha_2^{(0)} & \cdots & & \\ & \cdots & \cdots & \beta_{n-2}^{(0)} & \\ & & \beta_{n-2}^{(0)} & \alpha_{n-1}^{(0)} & \beta_{n-1}^{(0)} \\ & & & \beta_{n-1}^{(0)} & \alpha_n^{(0)} \end{pmatrix}.$$

One step of the implicitly shifted QR method can be summarized as follows. Let

$$G_1 = \begin{pmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{pmatrix}$$

be the Givens rotation such that $G_1[\alpha_1^{(0)} - \kappa, \beta_1^{(0)}]^T = [\tilde{\alpha}_1^{(0)}, 0]^T$, where κ is a

suitable shift and let

$$\hat{G}_1 = \begin{pmatrix} G_1 & \\ & I_{n-2} \end{pmatrix}.$$

The structure of the similar matrix $\hat{G}_1 T_n^{(0)} \hat{G}_1^T$ differs from a tridiagonal one for the same bulge, due to the symmetry, in position (1, 3) and (3, 1).

The rest of the algorithm consists of moving the bulges downwards and eventually remove them by using $n - 2$ more Givens similarity transformations \hat{G}_i , $i = 2, \dots, n - 1$, as depicted in Fig. 1. Then we pose

$$T_n^{(1)} = \hat{G}_{n-1} \cdots \hat{G}_1 T_n^{(0)} \hat{G}_1^T \cdots \hat{G}_{n-1}^T.$$

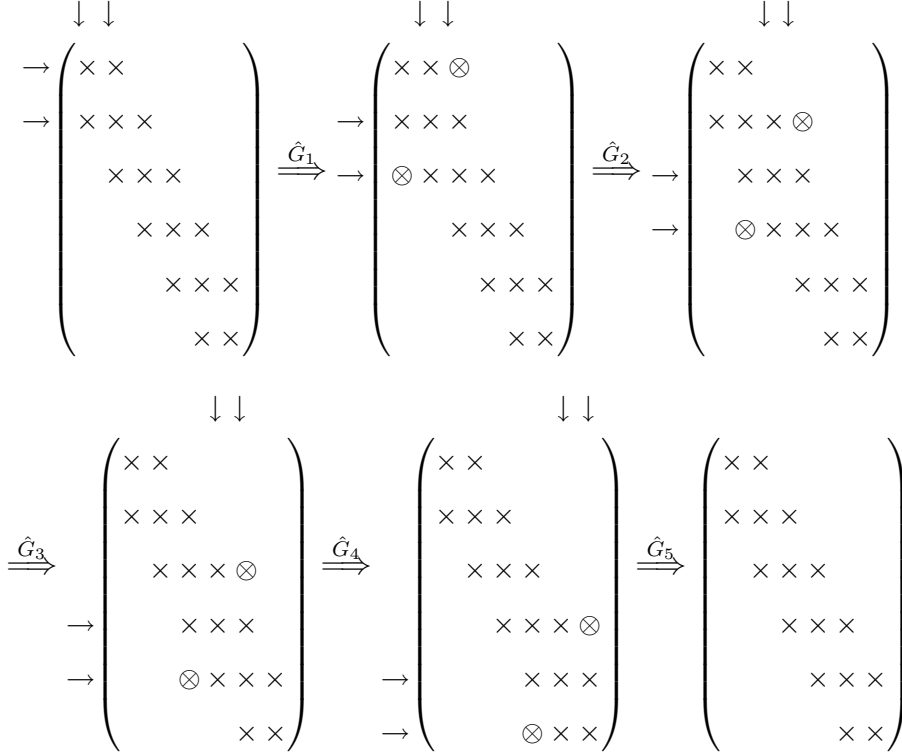


Fig. 1. One step of the implicitly shifted QR method applied to a tridiagonal matrix. The notation \hat{G}_j means that the similarity transformation $\hat{G}_j A \hat{G}_j^T$ is performed, with A the matrix to the left of the latter symbol.

The `matlab` function `QRstep` performing one step of the implicitly shifted QR -method on symmetric tridiagonal matrices can be found in the Appendix.

The orthogonal matrix generated by one step of the QR -method is an upper Hessenberg semiseparable matrix. For instance, if $n = 5$, the latter matrix

looks like

$$H_1(\kappa) = \begin{pmatrix} c_1 & -s_1c_2 & s_1s_2c_3 & -s_1s_2s_3c_4 & s_1s_2s_3s_4c_5 & -s_1s_2s_3s_4s_5 \\ s_1 & c_1c_2 & -c_1s_2c_3 & c_1s_2s_3c_4 & -c_1s_2s_3s_4c_5 & c_1s_2s_3s_4s_5 \\ & s_2 & c_2c_3 & -c_2s_3c_4 & c_2s_3s_4c_5 & -c_2s_3s_4s_5 \\ & & s_3 & c_3c_4 & -c_3s_4c_5 & c_3s_4s_5 \\ & & & s_4 & c_4c_5 & -c_4s_5 \\ & & & & s_5 & -c_5 \end{pmatrix}.$$

Exploiting the semiseparable structure of $H_1(\kappa)$ the product by a vector can be accomplished in $6n$ flops. Of course, $H_1(\kappa)$ does not need to be explicitly computed. It is sufficient to store the sine and cosine coefficients s_i and c_i , $i = 1, \dots, n-1$, into two vectors, respectively. The corresponding `matlab` function `prodgiv`, performing the product $H_1(\kappa)x$, with $x \in \mathbb{R}^n$, can be found in the Appendix.

From a theoretical point of view, it turns out that $\beta_{n-1}^{(1)} = 0$ if $\kappa \equiv \lambda_i$, i.e., if κ is one of the eigenvalues of $T_n^{(1)}$, also called *perfect shift* [9]. Moreover, the last column of the matrix $H_1(\lambda_i)$ is the normalized eigenvector of $T_n^{(0)}$ associated to λ_i . Therefore, known the eigenvalue λ_i of $T_n^{(0)}$, the corresponding eigenvector can be computed applying one step of the implicitly shifted *QR* method, choosing the shift equal to λ_i . Exploiting the semiseparable structure of $H_1(\lambda_i)$ the corresponding eigenvector can be computed in $2n$ floating point operations, by means of a simplified version of the function `prodgiv`. In fact, the last column of $H_1(\lambda_i)$ can be computed as the product of the latter matrix by e_n , the last vector of the canonical basis of \mathbb{R}^n . Therefore, only $2n$ flops are needed to compute the latter product, since the first $n-1$ entries of e_n are equal to zero.

Unfortunately, if the last entries of the eigenvector are very tiny, then $\beta_{n-1}^{(1)}$ could be not small [9] and the last column of $H_1(\lambda_i)$ could be not parallel to the eigenvector associated to λ_i .

In the next section we show how to overcome this drawback.

3 "Iterative refinement" for the computation of an eigenvector

In this section we show how to compute the eigenvector of $T_n^{(0)}$ associated to λ_i by using the implicitly shifted *QR* approach.

The idea is to iterate the *QR* method using λ_i as shift. Due to the cubic

convergence of the implicitly shifted QR method, after few iterations, let's say l iterations, $|\beta_{n-1}^{(l)}| < \mathbf{tol}$, with \mathbf{tol} the fixed tolerance. In general, \mathbf{tol} is chosen to be equal to $\varepsilon(|\alpha_{n-1}^{(l)}| + |\alpha_n^{(l)}|)$ [9], where ε is the machine precision. The corresponding eigenvector of $T_n^{(0)}$ associated to λ_i is given by

$$H_l(\lambda_i)H_{l-1}(\lambda_i) \cdots H_1(\lambda_i)e_n. \quad (1)$$

We observe that the product $H_l(\lambda_i)H_{l-1}(\lambda_i) \cdots H_1(\lambda_i)$ is not explicitly computed. In fact, it is only necessary to store the sine and cosine coefficients generated at each step of the method. Furthermore, exploiting the semiseparable structure of $H_j(\lambda_i)$, $j = 1, \dots, l$, the corresponding eigenvector is computed in $6ln$ floating point operations. On average, very few steps of iterative refinement are sufficient to compute the eigenvector up to the considered tolerance.

4 Numerical multiple eigenvalues

One of the most used methods to compute the eigenvectors of a tridiagonal matrix is the inverse iterations method. The blemish in this approach is that the computed eigenvectors may not be numerically orthogonal if clusters are present in the spectrum. To enforce orthogonality the Gram-Schmidt procedure is used, requiring $O(n^3)$ operations in the worst case. At first sight, also the proposed method of section 3 suffers from the loss of orthogonality in the presence of clusters in the spectrum. In this section we adapt the method of the previous section to overcome this drawback. Suppose the eigenvalues are ordered such that,

$$\lambda_i \leq \lambda_j, \quad i < j.$$

Two eigenvalues do not belong to the same cluster, if the following condition is satisfied [2].

Condition 4.1 *The eigenvalue λ should have an adequate separation from the rest of the spectrum, i.e.,*

$$\text{gap}(\lambda) := \min_{\lambda \neq \mu} |\lambda - \mu| \geq \mathbf{tolg},$$

where μ ranges over the other eigenvalues of $T_n^{(0)}$.

A good choice of \mathbf{tolg} is $10^{-3}\|T_n^{(0)}\|_\infty$, [2], [12, p. 322]. Therefore, we will say that two eigenvalues λ and μ of $T_n^{(0)}$ belong to the same cluster if

$$|\lambda - \mu| < \mathbf{tolg}.$$

Without loss of generality, we suppose that a cluster of 2 eigenvalues is present in the spectrum, i.e., $\lambda_i \approx \lambda_{i+1}$, $1 \leq i < n$.

The algorithm to compute the eigenvectors corresponding to the eigenvalues belonging to the cluster is depicted in Fig. 2. The algorithm can be easily extended to a cluster of larger size.

-
- (1) Iterate the QR method with implicit shift λ_{i+1} until $|\beta_j^{(l)}| < \text{tol}$, for some l and $n - 2 \leq j \leq n - 1$.
 - (2) Let $\mathbf{H}_l = H_l(\lambda_{i+1})H_{l-1}(\lambda_{i+1}) \cdots H_1(\lambda_{i+1})$.
 - (3) If $j = n - 1$, then $\mathbf{H}_l e_n$ is the eigenvector of $T_{n-1}^{(0)}$ corresponding to λ_{i+1} .
 - (a) Deflate $T_n^{(l)}$ (delete the last row and the last column).
 - (b) Let $T_{n-1}^{(0)} = T_{n-1}^{(l)}(1 : n - 1, 1 : n - 1)$ and iterate the QR -method with implicit shift λ_i until $|\beta_{n-2}^{(m)}| < \text{tol}$, for some m .
 - (c) Let $\hat{\mathbf{H}}_m = \tilde{H}_m(\lambda_i)\tilde{H}_{m-1}(\lambda_i) \cdots \tilde{H}_1(\lambda_i)$ and

$$\tilde{\mathbf{H}}_m = \begin{pmatrix} \hat{\mathbf{H}}_m \\ 1 \end{pmatrix},$$

where $\tilde{H}_k(\lambda_i)$ are the upper Hessenberg orthogonal semiseparable matrices generated by one step of the QR -method applied to $T_{n-1}^{(0)}$.

- (d) $\tilde{\mathbf{H}}_m \mathbf{H}_l e_{n-1}$ is the eigenvector of $T_{n-1}^{(0)}$ corresponding to λ_i .
- (4) Else (if $j = n - 2$)
 - (a) $\mathbf{H}_l(\lambda_{i+1})[e_{n-1}, e_n]$ is the subspace spanned by the eigenvectors of $T_n^{(0)}$ associated to the eigenvalues λ_i and λ_{i+1} .
 - (b) Let

$$\check{Q}^T \begin{pmatrix} \lambda_i \\ \lambda_{i+1} \end{pmatrix} \check{Q}$$

be the spectral decomposition of $T_n^{(0)}(n - 1 : n, n - 1 : n)$.

- (c) Let

$$\check{\mathbf{Q}} = \begin{pmatrix} I_{n-2} \\ \check{Q} \end{pmatrix}.$$

- (d) $\check{\mathbf{Q}}\mathbf{H}_l(\lambda_{i+1})e_{n-1}$ and $\check{\mathbf{Q}}\mathbf{H}_l(\lambda_{i+1})e_n$ are the eigenvectors of $T_n^{(0)}$ associated to the eigenvalues λ_i and λ_{i+1} .
-

Fig. 2. Algorithm for computing the eigenvectors corresponding to a cluster consisting of $\{\lambda_i, \lambda_{i+1}\}$.

Remark 1 *If the size of the cluster is k , approximately $O(k^2n)$ floating point operations are needed to compute the eigenvectors. In many cases, however, more than one entry in the subdiagonal of the tridiagonal matrix $T_n^{(l)}$ is tiny. Therefore, more than one row and column can be deflated from the matrix reducing considerably the computational complexity (see the numerical examples).*

5 Numerical examples

In this section, the proposed method is used to compute the eigenvectors of symmetric irreducible tridiagonal matrices, supposing the eigenvalues have been already computed in a certain way. There are many methods to compute the eigenvalues of a symmetric tridiagonal matrix [9,8]. In the following example we consider the values yielded by the function `eig` of `matlab` as the computed eigenvalues. The eigenvectors computed by the proposed method (denoted by `eigvp`) are compared with those computed by means of the `matlab` function `eig` (denoted by `eigvM`).

For each example, the maximum of the residuals,

$$\max_i \frac{\|T\hat{x}_i - \hat{\lambda}_i\hat{x}_i\|_2}{n\varepsilon\|T\|_2}$$

and the orthogonality factors

$$\max_i \frac{\|\hat{X}^T\hat{x}_i - e_i\|_2}{n\varepsilon}$$

are computed in order to measure the quality of the computed eigenvectors by the proposed method. In the following tables, n and $\#It$ denote the order of the considered matrices and the average number of iterations of the implicitly shifted QR method to compute one eigenvector by the proposed algorithm, respectively.

The first four examples can be found in [7]. In the last two examples, the well-known Wilkinson matrices and the glued Wilkinson matrices, for different values of n , are considered.

Given the eigenvalues λ_i , $i = 1, \dots, n$, an orthogonal matrix Q of the QR factorization of a random matrix of order n generated by the `matlab` function `randn` is computed. Thus, the matrix $T_n^{(0)}$ is obtained applying the Householder reduction into tridiagonal form to the matrix $Q\text{diag}(\lambda_1, \dots, \lambda_n)Q^T$. All the matrices in the examples are computed in this way, unless otherwise specified.

We can see that the eigenvectors computed by the proposed method have a behavior similar to those of the eigenvectors computed by the `matlab` function `eig`.

Example 5.1 *For this example, the eigenvalues are*

$$\lambda_i = \varepsilon + (i - 1)\tau, \quad i = 1, \dots, n, \quad \text{with } \tau = \frac{1 - \varepsilon}{n - 1}.$$

The eigenvalues are well separated and no clusters are present in the spectrum.

The numerical results are shown in table 1. The averaged number of iterations of the implicitly shifted QR method is very low and it slightly increases with the size of the considered matrices.

n	#It.	$\max_i \frac{\ T\hat{x}_i - \hat{\lambda}_i \hat{x}_i\ _2}{n\varepsilon\ T\ _2}$		$\max_i \frac{\ \hat{X}^T \hat{x}_i - e_i\ _2}{n\varepsilon}$	
		eigv _p	eigv _M	eigv _p	eigv _M
50	2.02	9.3030e-02	1.1019e-01	1.0449e+00	3.3765e-01
100	2.24	6.1713e-02	7.6537e-02	2.3864e+00	2.8038e-01
150	2.57	2.0597e-02	4.3215e-02	2.4162e+00	1.3883e-01
200	2.84	4.5924e-02	6.2735e-02	2.8777e+00	2.1381e-01
250	3.11	1.5222e-02	3.8523e-02	3.2206e+00	1.2914e-01

Table 1

Example 5.2 The eigenvalues of this example are

$$\lambda_i = \pm(\varepsilon + (i - 1)\tau), \quad i = 1, \dots, n, \quad \text{with } \tau = \frac{1 - \varepsilon}{n - 1},$$

i.e., the same as the latter one, with the sign chosen randomly. Also in this case no clusters are present in the spectrum. As expected, the results are very similar to those of the previous example, and are depicted in table 2.

n	#It.	$\max_i \frac{\ T\hat{x}_i - \hat{\lambda}_i \hat{x}_i\ _2}{n\varepsilon\ T\ _2}$		$\max_i \frac{\ \hat{X}^T \hat{x}_i - e_i\ _2}{n\varepsilon}$	
		eigv _p	eigv _M	eigv _p	eigv _M
50	2.10	9.1943e-02	1.3489e-01	5.9190e-01	8.3252e-02
100	2.38	6.7323e-02	1.0893e-01	6.3694e-01	6.7644e-02
150	2.59	9.7184e-02	1.1285e-01	7.0963e-01	5.9745e-02
200	2.94	2.7929e-02	7.4577e-02	7.1364e-01	4.1609e-02
250	3.22	5.2895e-02	1.1169e-01	7.8419e-01	4.0424e-02

Table 2

Example 5.3 The eigenvalues of the matrices constructed in this example are

$$\lambda_i = i \cdot \varepsilon, \quad i = 1, 2, \dots, n - 1, \quad \text{and } \lambda_n = 1.$$

There is a simple eigenvalue λ_n and the cluster $\{\lambda_i\}_{i=1}^{n-1}$ of size $n - 1$. Also in this case the cost of computing the whole set of eigenvectors is $O(n^2)$ because,

after very few iterations, the norm of the whole subdiagonal of the tridiagonal matrices generated by the QR method is very tiny.

n	#It.	$\max_i \frac{\ T\hat{x}_i - \lambda_i \hat{x}_i\ _2}{n\varepsilon \ T\ _2}$		$\max_i \frac{\ \hat{X}^T \hat{x}_i - e_i\ _2}{n\varepsilon}$	
		eigv _p	eigv _M	eigv _p	eigv _M
50	1.88	5.1777e-02	2.0000e-02	2.0073e-01	2.7508e-01
100	1.90	4.9323e-02	1.5013e-02	2.8044e-01	1.9137e-01
150	1.94	4.7127e-03	1.7893e-03	3.7348e-01	1.9556e-01
200	1.94	5.1156e-03	2.5195e-03	4.6506e-01	1.9910e-01
250	1.98	5.0025e-03	2.0266e-03	6.8008e-01	1.3937e-01

Table 3

Example 5.4 In this example the Toeplitz matrices of order n are considered,

$$T_n^{(0)} = \begin{pmatrix} 2 & 1 & & & \\ & 1 & 2 & \cdots & \\ & & \cdots & \cdots & 1 \\ & & & & 1 & 2 \end{pmatrix}.$$

Its eigenvalues are

$$\lambda_i = 4 \sin^2 \frac{i\pi}{2(n+1)}, \quad i = 1, \dots, n,$$

and are well separated from each other except those in the neighbourhood of 0 and 4. The results are depicted in table 4.

n	#It.	$\max_i \frac{\ T\hat{x}_i - \lambda_i \hat{x}_i\ _2}{n\varepsilon \ T\ _2}$		$\max_i \frac{\ \hat{X}^T \hat{x}_i - e_i\ _2}{n\varepsilon}$	
		eigv _p	eigv _M	eigv _p	eigv _M
50	1.92	5.2274e-02	1.0127e-01	5.2206e-01	1.0500e-01
100	1.94	3.0002e-02	5.7680e-02	4.7589e-01	6.0098e-02
150	1.98	2.6499e-02	4.5126e-02	8.4123e-01	4.5312e-02
200	2.01	2.7503e-02	3.9267e-02	7.0283e-01	4.3592e-02
250	2.03	1.8502e-02	3.5513e-02	8.6697e-01	3.2414e-02

Table 4

sidered,

$$W_{21,p}^+ = \begin{pmatrix} W_{21}^+ & M & & & \\ M^T & W_{21}^+ & M & & \\ & \ddots & \ddots & \ddots & \\ & & M^T & W_{21}^+ & M \\ & & & M^T & W_{21}^+ \end{pmatrix}, \text{ where } M = 10^{-12}e_{21}e_1^T.$$

The results are reported in table 6.

n	#It.	$\max_i \frac{\ T\hat{x}_i - \hat{\lambda}_i \hat{x}_i\ _2}{n\varepsilon \ T\ _2}$		$\max_i \frac{\ \hat{X}^T \hat{x}_i - e_i\ _2}{n\varepsilon}$	
		eigv _p	eigv _M	eigv _p	eigv _M
42	1.45	4.2113e-01	9.9321e-01	2.2523e+01	6.3716e+00
105	1.45	8.0750e-01	9.2957e-01	1.8972e+01	4.2258e+00
210	1.45	7.8312e-01	1.1324e+00	1.0983e+01	3.4728e+00
315	1.47	4.8794e-01	7.5062e-01	9.0341e+00	2.4917e+00
420	1.54	4.8022e-01	7.6561e-01	9.9453e+00	2.4357e+00
525	1.57	3.4735e-01	9.0144e-01	4.7771e+00	2.1117e+00

Table 6

6 Conclusions

A numerical method for computing the eigenvectors of symmetric tridiagonal matrices is proposed in this paper. This method can easily be adapted for other classes of matrices, e.g. semiseparable matrices, as long as a step of the QR method requires $O(n)$ floating point operations. The eigenvectors are numerically orthogonal. They are computed in $O(n^2)$ floating point operations, where n is the order of the matrix if few clusters of small size are present in the spectrum. Even in presence of clusters of large size, the algorithm often computes the eigenvectors with $O(n^2)$ floating point operations, because more than one eigenvector is computed at each iteration of the method.

References

- [1] A. DAX, *The Orthogonal Rayleigh Quotient Iteration (ORQI) method*, Linear Algebra and its Applications, vol. 358:1-3, pp. 23-43, 2003.
- [2] I. S. DHILLON AND A. N. MALYSHEV, *Inner deflation of symmetric, tridiagonal matrices*, Linear Algebra and its Applications, vol. 358:1-3, pp. 139-144, 2003.
- [3] I. S. DHILLON, *Current inverse iteration software can fail*, BIT, vol. 38:4, pp. 685–704, 1998.
- [4] I. S. DHILLON AND B. N. PARLETT, *Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices*, Linear Algebra and its Applications, vol. 387, pp. 1-28, 2004.
- [5] I. S. DHILLON AND B. N. PARLETT, *Orthogonal eigenvectors and relative gaps*, SIAM Journal on Matrix Analysis and Applications, vol. 25:3, pp. 858–899, 2004.
- [6] B. N. PARLETT AND I. S. DHILLON, *Fernando’s solution to Wilkinson’s problem: an application of double factorization*, Linear Algebra and its Applications, vol. 267, pp. 247–279, 1997.
- [7] I. S. DHILLON, *A new $O(N^2)$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem*, PhD. Thesis, University of California, Berkeley, 1997.
- [8] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Second ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
- [9] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Classic ed., SIAM, Philadelphia, PA, 1998.
- [10] M. VAN BAREL, R. VANDEBRIL, AND N. MASTRONARDI, *The Lanczos-Ritz values appearing in an orthogonal similarity reduction of a matrix into semiseparable form*, Report TW 360, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 2003, accepted for publication in SIAM J. Matrix Anal. Appl.
- [11] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *An implicit QR algorithm for semiseparable matrices to compute the eigendecomposition of symmetric matrices*, Report TW 367, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, August 2003.
- [12] J.H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University press, Oxford, 1965.

7 Appendix

7.1 function QRstep

```
function [ $\alpha$ ,  $\beta$ ,  $C$ ,  $S$ ] = QRstep ( $\alpha$ ,  $\beta$ ,  $\kappa$ ,  $n$ );
% Input:    $\alpha$  - the main diagonal of the symmetric tridiagonal matrix
%           $\beta$  - the subdiagonal of the symmetric tridiagonal matrix
%           $\kappa$  - the shift used in the current step
%           $n$  - the order of the matrix
% Output:   $\alpha$  - the new main diagonal of the symmetric tridiagonal matrix
%           $\beta$  - the new subdiagonal of the symmetric tridiagonal matrix
%           $C$  - the vector of the cosines coefficients of the Givens rotations
%           $S$  - the vector of the sines coefficients of the Givens rotations
% one step of implicitly shifted QR algorithm;
 $g = \alpha(1) - \kappa$ ;
 $s = 1$ ;  $c = 1$ ;  $p = 0$ ;
for  $i = 1 : n - 1$ ,
     $f = s \times \beta(i)$ ;
     $b = c \times \beta(i)$ ;
    [ $c$ ,  $s$ ,  $g$ ] = Givens( $g$ ,  $f$ );
     $C(i) = c$ ;  $S(i) = s$ ;
    if  $i \sim 1$ ,
         $\beta(i - 1) = g$ ;
    end
     $u = \alpha(i) - p$ ;
     $v = (\alpha(i + 1) - u) \times s + 2 \times c \times b$ ;
     $p = s \times v$ ;
     $\alpha(i) = u + p$ ;
     $g = c \times v - b$ ;
end
 $\alpha(k) = \alpha(k) - p$ ;
 $\beta(k - 1) = g$ ;
```

Computational complexity: $12n$ floating point operations plus $7n$ floating point operations due to the computation of the Givens coefficients by the function `Givens` [8].

7.2 *function* prodgiv

```
function[x] =prodgiv (C, S, x, n);
% Input:   C   - the vector of the cosines coefficients of the Givens rotations
%          S   - the vector of the sines coefficients of the Givens rotations
%          x   - a vector of length n
% Output:  x   - the vector of the product  $H_1(\kappa)x$ 
% fast product of  $n - 1$  Givens times a vector  $x$ 
for i = n - 1 : -1 : 1,
    t = C(i) × x(i) - S(i) × x(i + 1);
    x(i + 1) = s(i) × x(i) + c(i) × x(i + 1);
    x(i) = t;
end
```

Computational complexity: $6n$ floating point operations.