# On Computing Eigenvectors of Symmetric Tridiagonal Matrices

**Nicola Mastronardi, Harold Taeter, and Paul Van Dooren**

**Abstract** The computation of the eigenvalue decomposition of symmetric matrices is one of the most investigated problems in numerical linear algebra. For a matrix of moderate size, the customary procedure is to reduce it to a symmetric tridiagonal one by means of an orthogonal similarity transformation and then compute the eigendecomposition of the tridiagonal matrix.

Recently, Malyshev and Dhillon have proposed an algorithm for deflating the tridiagonal matrix, once an eigenvalue has been computed. Starting from the aforementioned algorithm, in this manuscript we develop a procedure for computing an eigenvector of a symmetric tridiagonal matrix, once its associate eigenvalue is known.

We illustrate the behavior of the proposed method with a number of numerical examples.

**Keywords** Tridiagonal matrices · Eigenvalue computation · $QR$ method

The author “Nicola Mastronardi” is a member of the INdAM Research group GNCS.
The scientific responsibility rests with its authors.

N. Mastronardi (✉)
Istituto per le Applicazioni del Calcolo “M. Picone”, Consiglio Nazionale delle Ricerche, sede di Bari, Italy
e-mail: n.mastronardi@ba.iac.cnr.it

H. Taeter
Dipartimento di matematica, Università degli Studi di Bari, Bari, Italy
e-mail: harold.taeter@uniba.it

P. Van Dooren
Department of Mathematical Engineering, Catholic University of Louvain, Louvain-la-Neuve, Belgium
e-mail: paul.vandooren@uclouvain.be

## 1 Introduction

Computing the eigenvalue decomposition of symmetric matrices is one of the most investigated problems in numerical linear algebra [6, 11]. For a matrix of moderate size, having reduced the symmetric matrix into a symmetric tridiagonal one by means of a similarity orthogonal transformation, the problem reduces to the computation of the eigendecomposition of a tridiagonal matrix.

There are different methods to compute the eigenvalues of symmetric tridiagonal matrices, such as the bisection method [14], the $QR$ method [14] and divide & conquer methods [2, 7]. For computing the eigenvectors one can use inverse iteration [14], the $QR$ method [14] and the multiple relatively robust representations algorithm [5, 13, 15]. The latter algorithm is based on the twisted factorization of the involved tridiagonal matrix to determine the position where the sought eigenvector has a large entry [5, 15, 16].

Once an eigenvalue is computed, a deflation algorithm was proposed in [4] in order to remove it from the tridiagonal matrix and reduce the dimension of the problem by one. Such an algorithm can also be used to compute the eigenvector associated to the computed eigenvalue and it is based on the twisted factorization used in [5, 15].

In this manuscript we consider a modified version of the aforementioned algorithm to compute an eigenvector of a symmetric tridiagonal matrix, supposing the corresponding eigenvalue is known.

Without loss of generality, we consider only the real case. The complex Hermitian one can be handled in the same way.

We illustrate the behavior of the proposed method with some numerical examples. The manuscript is organized as follows. In Sect. 2 the notation used in the manuscript is given. In Sect. 3 the main features of the $QR$ method are described. The proposed algorithm is described in Sect. 4, followed by the section of numerical examples and by the conclusions.

## 2 Notations and Definitions

Matrices are denoted with upper case letters and their entries with lower case letters, i.e., the element $(i, j)$ of the matrix $T$ is denoted by $t_{i,j}$.

The submatrix of the matrix $B$ made by the rows $i, i+1, i+2, \ldots, i+k$, with $1 \leq i \leq n-k$, $0 \leq k \leq n-i$, and columns $j, j+1, j+2, \ldots, j+l$, with $1 \leq j \leq n-l$, $0 \leq l \leq n-j$, is denoted by $B_{i:i+k,j:j+l}$. If the matrix $T$ is symmetric, the submatrix made by the rows and columns $i, i+1, i+2, \ldots, i+k$, with $1 \leq i \leq n-k$, $0 \leq k \leq n-i$, is simply denoted by $T_{i:i+k}$.

The identity matrix of order $n$ is denoted by $I_n$ or by $I$ if there is no ambiguity.

The matrix $T - \kappa I$, with $\kappa \in \mathbb{R}$, is denoted by $T(\kappa)$.

The principal diagonal of a matrix $B \in \mathbb{R}^{m \times n}$ is denoted by $\text{diag}(B)$.

The machine precision is denoted by $\varepsilon$.

The $i$th vector of the canonical basis of $\mathbb{R}^n$ is denoted by $\mathbf{e}_i^{(n)}$, or simply by $\mathbf{e}_i$, if there is no ambiguity.

**Definition 1** Given $B \in \mathbb{R}^{m\times n}, m \geq n$, let $B = U\Sigma V^T$ be its singular value decomposition, with $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ orthogonal and $\Sigma \in \mathbb{R}^{m\times n}$ diagonal, with $\mathrm{diag}(\Sigma) = \left[\sigma_1, \sigma_2, \cdots \sigma_n\right]^T$, and $\sigma_i \geq \sigma_{i+1},\ i = 1,\ldots,n-1$.

The columns of $B$ are said $\varepsilon$-linear dependent if $\sigma_n \leq \varepsilon\|B\|_2$.

The columns of $B$ are said *strongly* linear independent if $\sigma_n \gg \varepsilon\|B\|_2 > 0$.

## 3 Implicit $QR$ Method

Let $T \in \mathbb{R}^{n\times n}$ be the symmetric tridiagonal matrix

$$T = \begin{bmatrix} t_{1,1} & t_{1,2} & & \\ t_{2,1} & t_{2,2} & \ddots & \\ & \ddots & \ddots & t_{n-1,n} \\ & & t_{n,n-1} & t_{n,n} \end{bmatrix},$$

with $t_{i,i+1} = t_{i+1,i},\ i = 1,\ldots,n-1$.

Let us suppose that $T$ is irreducible, i.e., $t_{i,i+1} \neq 0,\ i = 1,\ldots,n-1$ and let $T = X\Lambda X^T$ be its eigenvalue decomposition, with $X \in \mathbb{R}^{n\times n}$ orthogonal, and $\Lambda \in \mathbb{R}^{n\times n}$ diagonal, with $\mathrm{diag}(\Lambda) = [\lambda_1,\ldots,\lambda_n]^T$. Since $T$ is irreducible, then $\lambda_i \neq \lambda_j$, with $i \neq j,\ i,j = 1,\ldots,n$.

The Implicit $QR$ (IQR) method is the standard method for computing the eigenvalue decomposition of matrices of moderate size [6]. In particular, `MATLAB` uses the `LAPACK` routine `DSYEV`, based on the $QR$ method, to compute eigenvalues and eigenvectors of a real symmetric matrix [1].

Given a symmetric irreducible tridiagonal matrix $T \in \mathbb{R}^{n\times n}$, and $\kappa \in \mathbb{R}$, one sweep of IQR with shift $\kappa$ consists of computing the similarity transformation

$$\hat{T}^{(n)} = \hat{G}_{n-1}\hat{G}_{n-2}\cdots\hat{G}_1\hat{T}^{(1)}\hat{G}_1^T\cdots\hat{G}_{n-2}^T\hat{G}_{n-1}^T,$$

where $\hat{T}^{(1)} = T$ and $\hat{G}_i,\ i = 1,\ldots,n-1$, are Givens rotations

$$\hat{G}_i = \begin{bmatrix} I_{i-1} & & & \\ & \hat{c}_i & \hat{s}_i & \\ & -\hat{s}_i & \hat{c}_i & \\ & & & I_{n-i-1} \end{bmatrix}, \quad i = 1,\ldots,n-1.$$

with $\hat{c}_i^2 + \hat{s}_i^2 = 1$. Without loss of generality, we assume that $\hat{c}_i \geq 0$. Hence the matrix $\hat{Q}$ in (1) is uniquely defined.

In particular, $\hat{G}_1$ is the Givens rotation acting on the first two rows of $\hat{T}^{(1)}$, whose coefficients $\hat{c}_1$ and $\hat{s}_1$ are such that

$$\begin{bmatrix} \hat{c}_1 & \hat{s}_1 \\ -\hat{s}_1 & \hat{c}_1 \end{bmatrix} \begin{bmatrix} \hat{t}^{(1)}_{1,1} - \kappa \\ \hat{t}^{(1)}_{2,1} \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_1 \\ 0 \end{bmatrix},$$

with $\hat{\alpha}_1 = \| \begin{bmatrix} \hat{t}^{(1)}_{1,1} - \kappa, \; \hat{t}^{(1)}_{2,1} \end{bmatrix} \|_2$. The structure of the matrix $\hat{T}^{(2)} = \hat{G}_1 \hat{T}^{(1)} \hat{G}_1^T$ differs from the one of a tridiagonal matrix for an entry different from 0 in position (3, 1) (and, symmetrically, in position (1, 3)), called "*bulge*".

Each of the other Givens rotations $\hat{G}_i$ are applied to move the bulge one position downward along the second subdiagonal/superdiagonal and eventually remove it [11], i.e, the matrix

$$\hat{T}^{(i)} = \hat{G}_{i-1}\hat{G}_{i-2}\cdots\hat{G}_1\hat{T}^{(1)}\hat{G}_1^T\cdots\hat{G}_{i-2}^T\hat{G}_{i-1}^T,$$

has the bulge in position $(i-1, i+1)$ (and, symmetrically, in position $(i+1, i-1)$), $\hat{T}^{(i+1)} = \hat{G}_i\hat{T}^{(i)}\hat{G}_i^T$ has the bulge in position $(i, i+2)$ (and, symmetrically, in position $(i+2, i)$), and so on. The matrix

$$\hat{Q} = \hat{G}_{n-1}\hat{G}_{n-2}\cdots\hat{G}_1 \tag{1}$$

is orthogonal Hessenberg.

In the sequel, we call the sweep of the IQR method described above a "*forward*" IQR (FIQR) sweep because it starts from the top-left corner of $\hat{T}_1$ and ends in the bottom-right one.

The IQR method can also be implemented in a "*backward*" fashion, i.e., starting from the bottom-right corner of $T$ and ending in the top-left corner [9]. We will refer to one sweep of this procedure as a backward IQR (BIQR) sweep.

Let $\tilde{T}^{(1)} = T$. In a BIQR sweep with shift $\kappa$, a sequence of Givens rotations

$$\tilde{G}_i = \begin{bmatrix} I_{n-i-1} & & & \\ & \tilde{c}_i & \tilde{s}_i & \\ & -\tilde{s}_i & \tilde{c}_i & \\ & & & I_{i-1} \end{bmatrix}, \quad i = 1, \ldots, n-1,$$

with $\tilde{c}_i^2 + \tilde{s}_i^2 = 1$, is determined in the following way.

The coefficients $\tilde{c}_1$ and $\tilde{s}_1$ of $\tilde{G}_1$ are computed such that

$$\begin{bmatrix} \tilde{t}^{(1)}_{n,n-1}, \; \tilde{t}^{(1)}_{n,n} - \kappa \end{bmatrix} \begin{bmatrix} \tilde{c}_1 & \tilde{s}_1 \\ -\tilde{s}_1 & \tilde{c}_1 \end{bmatrix} = \begin{bmatrix} 0, \; \tilde{\alpha}_n \end{bmatrix},$$

with $\tilde{\alpha}_n = \| \begin{bmatrix} \tilde{t}^{(1)}_{n,n-1}, \; \tilde{t}^{(1)}_{n,n} - \kappa \end{bmatrix} \|_2$. The matrix $\tilde{T}^{(2)} = \tilde{G}_1^T\tilde{T}^{(1)}\tilde{G}_1$ has a bulge in position $(n, n-2)$ (and, symmetrically, in position $(n-2, n)$).

The Givens rotations $\tilde{G}_i,\ i = 2, \ldots, n-1$, are sequentially applied to $\tilde{T}^{(2)}$ to move the bulge upward along the second subdiagonal and eventually remove it in the matrix $\tilde{T}^{(n)} = \tilde{G}_{n-1}^T \tilde{G}_{n-2}^T \cdots \tilde{G}_1^T \tilde{T}^{(1)} \tilde{G}_1 \cdots \tilde{G}_{n-2} \tilde{G}_{n-1}$.

Let $\tilde{Q} = \tilde{G}_1 \cdots \tilde{G}_{n-2} \tilde{G}_{n-1}$. Without loss of generality, we assume $\tilde{c}_i \geq 0$, which makes the matrix $\tilde{Q}$ uniquely defined.

Let $\lambda$ be an eigenvalue of $T$ with corresponding eigenvector $\mathbf{x}$. In infinite precision arithmetic, if $\lambda$ is chosen as shift $\kappa$ in the FIQR sweep, $\lambda$ shows up in position $(n, n)$ of $\hat{T}^{(n)}$. Moreover, $\hat{t}_{n-1,n}^{(n)} = \hat{t}_{n,n-1}^{(n)} = 0$, and $\mathbf{x} = \hat{Q}(:, n)$. In particular, since

$$\hat{Q} =$$
$$\begin{bmatrix}
\hat{c}_1 & -\hat{s}_1\hat{c}_2 & \hat{s}_1\hat{s}_2\hat{c}_3 & \ddots & -1^n \hat{c}_{n-1} \prod_{i=1}^{n-2} \hat{s}_i & -1^{n+1} \prod_{i=1}^{n-1} \hat{s}_i \\
\hat{s}_1 & \hat{c}_1\hat{c}_2 & -\hat{c}_1\hat{s}_2\hat{c}_3 & \ddots & -1^{n-1}\hat{c}_1\hat{c}_{n-1} \prod_{i=2}^{n-2} \hat{s}_i & -1^n \hat{c}_1 \prod_{i=2}^{n-1} \hat{s}_i \\
 & \hat{s}_1 & \hat{c}_1\hat{c}_2 & \ddots & \vdots & \vdots \\
 & & \ddots & \ddots & -\hat{c}_{n-3}\hat{s}_{n-2}\hat{c}_{n-1} & \hat{c}_{n-3}\hat{s}_{n-2}\hat{s}_{n-1} \\
 & & & \hat{s}_{n-2} & \hat{c}_{n-2}\hat{c}_{n-1} & -\hat{c}_{n-2}\hat{s}_{n-1} \\
 & & & & \hat{s}_{n-1} & \hat{c}_{n-1}
\end{bmatrix},$$

then

$$\mathbf{x} = \hat{Q}(:, n) = \begin{bmatrix}
-1^{n+1} \prod_{i=1}^{n-1} \hat{s}_i \\
-1^n \hat{c}_1 \prod_{i=2}^{n-1} \hat{s}_i \\
-1^{n-1} \hat{c}_2 \prod_{i=3}^{n-1} \hat{s}_i \\
\vdots \\
\hat{c}_{n-3}\hat{s}_{n-2}\hat{s}_{n-1} \\
-\hat{c}_{n-2}\hat{s}_{n-1} \\
\hat{c}_{n-1}
\end{bmatrix}. \tag{2}$$

Analogously, in infinite precision arithmetic, if $\lambda$ is chosen as shift $\kappa$ in the BIQR sweep, $\lambda$ shows up in position (1, 1) of $\tilde{T}^{(n)}$. Moreover, $\tilde{t}_{1,2}^{(n)} = \tilde{t}_{2,1}^{(n)} = 0$, and $\mathbf{x} = \tilde{Q}(1, :)^T$,

$$\mathbf{x} = \begin{bmatrix}
\tilde{c}_{n-1} \\
-\tilde{c}_{n-2}\tilde{s}_{n-1} \\
\tilde{c}_{n-3}\tilde{s}_{n-2}\tilde{s}_{n-1} \\
\vdots \\
-1^{n-1} \tilde{c}_2 \prod_{i=3}^{n-1} \tilde{s}_i \\
-1^n \tilde{c}_1 \prod_{i=2}^{n-1} \tilde{s}_i \\
-1^{n+1} \prod_{i=1}^{n-1} \tilde{s}_i
\end{bmatrix}. \tag{3}$$

Therefore, for a given eigenvalue $\lambda$, it is suggested in [11] to apply one sweep of either forward or backward IQR with shift $\lambda$ to compute the corresponding eigenvector with $O(n)$ floating point operations [11].

Unfortunately, forward instability can occur in floating point arithmetic in one forward/backward IQR sweep with shift $\lambda$ and the last column of $\hat{Q}$ (the first row of $\tilde{Q}$) may be far from the sought eigenvector [12].

In particular, forward instability occurs at step $j$ of one sweep of FIQR if and only if the shift $\kappa$ is very close to one of the eigenvalues of $\hat{T}^{(j)}_{1:j,1:j}$ and the last entry of the corresponding eigenvector is tiny [12]. As a consequence, the entries $t^{(j)}_{j,j-1}$ and $t^{(j)}_{j,j+1}$ are *"sufficiently"* small[1] [12]. By (2), the last component of the eigenvector is given by $\hat{c}_j$. Hence, forward instability happens if $\kappa$ is very close to one of the eigenvalues of $\hat{T}_{1:j,1:j}$ and $\hat{c}_j \sim O(\varepsilon)$. This means that the first $j$ columns of $\hat{T}_{1:j,1:j}$ are $\varepsilon$-linear dependent.

The same phenomenon can occur in a BIQR sweep.

To examine in which step of a IQR sweep forward instability can occur, let us consider the following Corollary [8, p.149].

**Corollary 1** *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix and let $B$ be a submatrix obtained by deleting $r$ rows from $A$. Then*

$$\sigma_k(A) \geq \sigma_k(B) \geq \sigma_{k+r}(A), \quad k = 1, \ldots, n,$$

*where $\sigma_\ell(A) \equiv 0, \quad if \;\; \ell > n.$*

Let us suppose that $\sigma_{n-1}(T(\lambda)) \gg \varepsilon > \sigma_n(T(\lambda)) = 0$ and $\sigma_{j-1}(T_{1:j,:}(\lambda)) \gg \sigma_j(T_{1:j,:}(\lambda)) \sim O(\varepsilon)$, for a $j \in \{2, \ldots, n\}$. By Corollary 1, all the submatrices $T_{1:j+\ell,:}(\lambda)$ are $\varepsilon$-singular, $\ell = 1, \ldots, n-j$, and

$$\sigma_{j+\ell}(T_{1:j+\ell,:}(\lambda)) \geq \sigma_{j+\ell}(T_{1:j+\ell,1:j+\ell}(\lambda)), \quad \ell = 1, \ldots, n-j,$$

i.e., the submatrices $T_{1:j+\ell,1:j+\ell}(\lambda)$ are $\varepsilon$-singular as well.

On the other hand,

$$\begin{gathered}\sigma_{j-1}(T_{n-j+1:n,:}(\lambda)) \geq \sigma_{n-1}(T(\lambda)) \gg \varepsilon, \\ \sigma_{j-1}(T_{n-j+1:n,:}(\lambda)) \geq \sigma_j(T_{n-j+1:n,n-j+1:n}(\lambda)) \geq \sigma_j(T_{n-j+1:n,:}(\lambda)).\end{gathered}$$

This means that forward instability is not encountered in the first $n-j-1$ steps of FIQR and in the first $j$ steps of BIQR.

In the following example it is shown how the sequences $\{\hat{c}_j\}_{j=1}^{n-1}$ and $\{\tilde{c}_j\}_{j=1}^{n-1}$, computed in floating point arithmetic, differ from those computed in infinite precision arithmetic.

[1]If one of the indices $i, j$ in $t_{i,j}$ is either 0 or $n$, we set $t_{i,j} \equiv 0$.

*Example 1* Let $T \in \mathbb{R}^{n\times n}$, $n = 100$, be a symmetric irreducible tridiagonal matrix with its entries generated by the `MATLAB` function `randn`.[2]

Let $T = X^{(M)}\Lambda^{(M)}X^{(M)^T}$ be the eigenvalue decomposition of $T$ computed by using the `MATLAB` function `eig`, with $\Lambda^{(M)} = \operatorname{diag}(\lambda_1^{(M)}\lambda_2^{(M)},\ldots,\lambda_n^{(M)})$, with $\lambda_i^{(M)} \ge \lambda_{i+1}^{(M)}$ $i = 1,\ldots,n-1$.

We report the behaviour of one sweep of F/B IQR with shift $\lambda_{19}^{(M)}$, although a similar behavior can be observed if we choose almost any $\lambda_i^{(M)}$, $i = 1,\ldots,n$, as a shift.

Let $(\bar\lambda, \bar{\mathbf{x}})$ be the eigenpair computed by a few steps of inverse iteration with initial guess $(\lambda_{19}^{(M)}, X^{(M)}(:,19))$. In this way, $\bar{\mathbf{x}}$ is computed with higher accuracy with respect to $X^{(M)}(:,19))$.

Let $\{\check c_i\}_{i=1}^{n-1}$ and $\{\bar c_i\}_{i=1}^{n-1}$ be the sequence of the cosines of the Givens matrices $\{\check G_i\}_{i=1}^{n-1}$ and $\{\bar G_i\}_{i=1}^{n-1}$, determined in order to transform $\bar{\mathbf{x}}$ to $\mathbf{e}_n$ and $\mathbf{e}_1$, respectively, i.e.,

$$\check G_i = \begin{bmatrix} I_{n-i-1} & & & \\ & \check c_i & \check s_i & \\ & -\check s_i & \check c_i & \\ & & & I_{i-1}\end{bmatrix}, \quad \text{such that} \quad \check G_{n-1}\check G_{n-2}\cdots\check G_1\bar{\mathbf{x}} = \mathbf{e}_n,$$

$$\bar G_i = \begin{bmatrix} I_{i-1} & & & \\ & \bar c_i & \bar s_i & \\ & -\bar s_i & \bar c_i & \\ & & & I_{n-i-1}\end{bmatrix}, \quad \text{such that} \quad \bar G_1\cdots\bar G_{n-2}\bar G_{n-1}\bar{\mathbf{x}} = \mathbf{e}_1.$$

Without loss of generality, we assume $\check c_i \ge 0$ and $\bar c_i \ge 0$, $i = 1,\ldots,n-1$.

Since $\bar{\mathbf{x}}$ is computed with high accuracy, the sequences $\{\check c_i\}_{i=1}^{n-1}$ and $\{\bar c_i\}_{i=1}^{n-1}$, are computed with high accuracy, too [10].

In infinite precision arithmetic, the sequences $\{\check c_i\}_{i=1}^{n-1}$ and $\{\hat c_i\}_{i=1}^{n-1}$ should be the same, while in floating point arithmetic the sequence $\{\hat c_i\}_{i=1}^{n-1}$ can depart from the sequence $\{\check c_i\}_{i=1}^{n-1}$ due to the forward instability [10]. The same holds for the sequences $\{\bar c_i\}_{i=1}^{n-1}$ and $\{\tilde c_i\}_{i=1}^{n-1}$.

The sequences $\{\hat c_i\}_{i=1}^{n-1}$, $\{\check c_i\}_{i=1}^{n-1}$, $\{|\hat t_{i-1,i}^{(i)}| + |\hat t_{i,i+1}^{(i)}|\}_{i=1}^{n-1}$, $\{\check\sigma_i\}_{i=1}^{n-1}$, and $\{\hat\sigma_i\}_{i=1}^{n-1}$, with $\check\sigma_i = \min(\operatorname{svd}(T_{:,i:n}(\lambda_{19}^{(M)})))$ and $\hat\sigma_i = \min(\operatorname{svd}(T_{i:n,i:n}(\lambda_{19}^{(M)})))$, denoted respectively by "$*$", "$+$", "$\circ$", "$\diamond$" and "$\nabla$", are displayed in Fig. 1 on a logarithmic scale.

We can observe that the first and the third sequence have a similar behaviour. The same can be said for the second and the fifth sequence. Moreover, the two sequences of cosines $\{\hat c_i\}_{i=1}^{n-1}$ and $\{\check c_i\}_{i=1}^{n-1}$ are similar until forward instability occurs, i.e., until $\hat c_i$ and $|\hat t_{i-1,i}^{(i)}| + |\hat t_{i,i+1}^{(i)}|$ are both greater than $O(\sqrt{\varepsilon})$.

[2]The matrix $T$ can be downloaded at users.ba.cnr.it/iac/irmanm21/TRID_SYM.

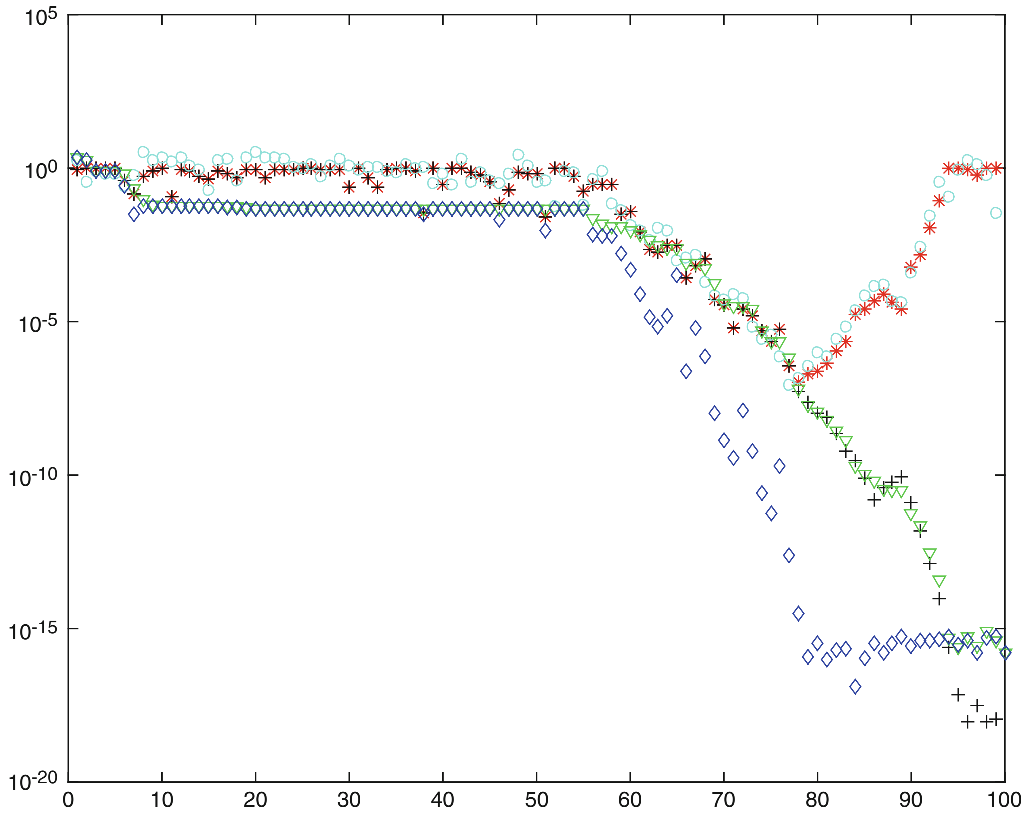

**Fig. 1** Sequences $\{\hat{c}_i\}_{i=1}^{n-1}$, $\{\check{c}_i\}_{i=1}^{n-1}$, $\{|\hat{t}_{i-1,i}^{(i)}| + |\hat{t}_{i,i+1}^{(i)}|\}_{i=1}^{n-1}$, $\{\check{\sigma}_i\}_{i=1}^{n-1}$, and $\{\hat{\sigma}_i\}_{i=1}^{n-1}$, denoted respectively by "asterisk", "plus", "circle", "diamond" and "triangledown", related to $T(\lambda_{19}^{(M)})$, with $T$ the matrix of Example 1 and $\lambda_{19}^{(M)}$ the 19th ordered eigenvalue computed by `eig` of `MATLAB`

The sequences $\{\tilde{c}_i\}_{i=1}^{n-1}$, $\{\bar{c}_i\}_{i=1}^{n-1}$, $\{|\tilde{t}_{n-i-1,n-i-2}^{(i)}| + |\tilde{t}_{n-i,n-i-1}^{(i)}|\}_{i=1}^{n-1}$, $\{\breve{\sigma}_i\}_{i=1}^{n-1}$, and $\{\hat{\sigma}_i\}_{i=1}^{n-1}$, with $\breve{\sigma}_i = \min(\text{svd}(T_{:,1:i}(\lambda_{19}^{(M)})))$, $\hat{\sigma}_i = \min(\text{svd}(T_{1:i,1:i}(\lambda_{19}^{(M)})))$, denoted respectively by "$*$", "$+$", "$\circ$", "$\diamond$" and "$\nabla$", are displayed in Fig. 2 in logarithmic scale.

Also in this case, the first and the third sequence have a similar behaviour and the same can be said for the second and the fifth sequence. Moreover, the two sequences of cosines $\{\tilde{c}_i\}_{i=1}^{n-1}$ and $\{\bar{c}_i\}_{i=1}^{n-1}$ are similar until forward instability occurs, i.e., until $\tilde{c}_i$ and $|\tilde{t}_{n-i-1,n-i-2}^{(i)}| + |\tilde{t}_{n-i,n-i-1}^{(i)}|$ are both greater than $O(\sqrt{\varepsilon})$.

Summarizing, forward instability occurs if the smallest singular value $\sigma_j^{(j)}$ of $T_{1:j,1:j}(\lambda)$ is close to the machine precision $\varepsilon$, for a certain $j \in \{1, \ldots, n\}$. As a consequence, the elements of the last column of $\hat{Q}$ of index greater than $j$ begin to depart from the elements of the eigenvector $\bar{\mathbf{x}}$. Moreover, $\hat{t}_{j,j-1}^{(j)} \approx \hat{t}_{j+1,j}^{(j)} \approx O(\sqrt{\varepsilon})$, where $\hat{t}_{i,k}^{(j)}$ is the $(i, k)$ entry of the matrix obtained after having applied $j$ Givens rotations in the forward IQR sweep with shift $\bar{\lambda}$ to $T_{1:n}(\bar{\lambda})$ [12]. The same holds to one sweep of BIQR, i.e., the first row of the upper Hessenberg matrix $\tilde{Q}$ is accurately

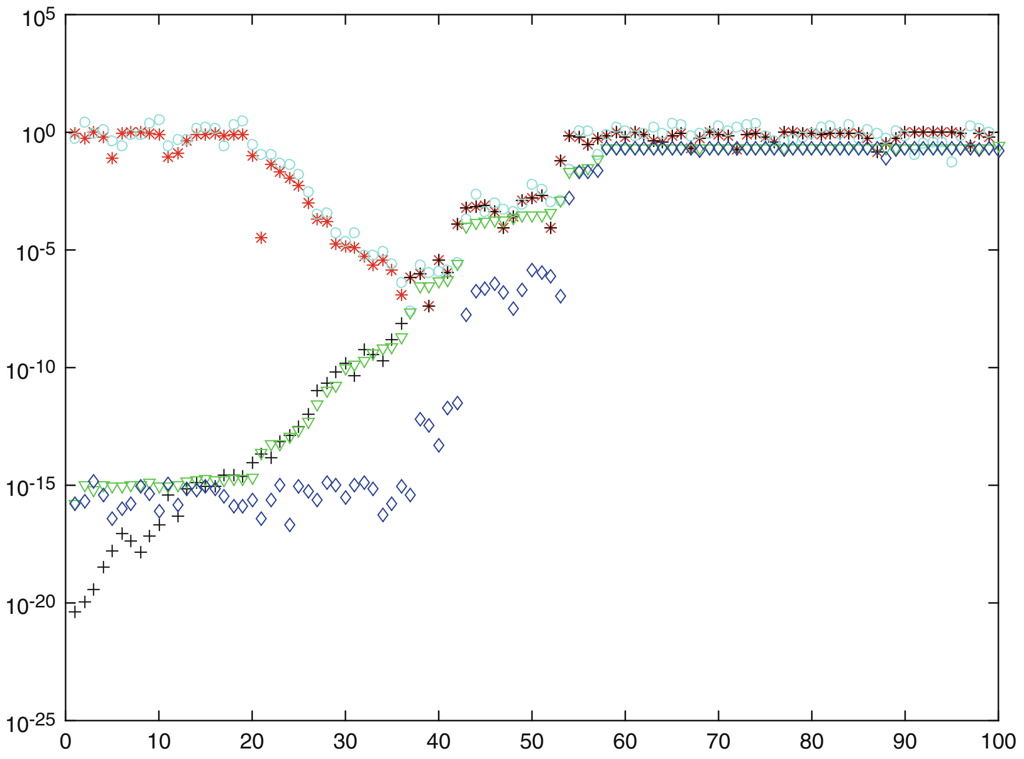

**Fig. 2** Sequences $\{\check{c}_i\}_{i=1}^{n-1}$, $\{\bar{c}_i\}_{i=1}^{n-1}$, $\{|\tilde{t}^{(i)}_{n-i-1,n-i-2}| + |\tilde{t}^{(i)}_{n-i,n-i-1}|\}_{i=1}^{n-1}$, $\{\bar{\sigma}_i\}_{i=1}^{n-1}$, and $\{\tilde{\sigma}_i\}_{i=1}^{n-1}$, denoted respectively by "asterisk", "plus", "circle", "diamond" and "triangledown", related to $T(\lambda_{19}^{(M)})$, with $T$ the matrix of Example 1 and $\lambda_{19}^{(M)}$ the 19th ordered eigenvalue computed by `eig` of `MATLAB`

computed as far as the smallest singular value of $T_{j:n}(\lambda)$ is large enough, for a certain $j \in \{1, \dots, n\}$.

Hence, the main issue is to determine the index $j$.

In the next section we consider the problem of determining the index $j$ such that the computed eigenvector will be obtained by $\hat{Q}(1 : j, n)$ and $\tilde{Q}(1, j+1 : n)^T$, i.e., gluing together the first $j$ entries of $\hat{Q}$ and the last $n - j$ entries of the first row of $\tilde{Q}$.

## 4 Computation of the Eigenvector

In this section we describe a technique to determine the index $j$ used for constructing the sought eigenvector by fetching the first $j$ entries of the last column of $\hat{Q}$ and the last $n - j$ entries of the first row of $\tilde{Q}$.

If $\sigma_{n-1}(T(\lambda)) \gg \sigma_n(T(\lambda))$ and forward instability occurs at step $j$ of one sweep of FIQR with shift $\lambda$, the sequence $\{\hat{c}_i\}_{i=1}^n$ begins to depart from the sequence $\{\check{c}_i\}_{i=1}^n$ around the index $j$. Analogously, the sequence $\{\tilde{c}_i\}_{i=1}^n$ begins to depart from the sequence $\{\bar{c}_i\}_{i=1}^{n}$ around the index $n-j$. Therefore, the sought index $j$ can be computed in the following way.

The sequence $\{\hat{c}_i\}_{i=1}^{n-1}$ generated by one FIQR sweep, is computed until $\hat{c}_{\hat{j}} < tol_1$ and $|\hat{t}^{(\hat{j})}_{\hat{j}-1,\hat{j}}|+|\hat{t}^{(\hat{j})}_{\hat{j},\hat{j}+1}| < tol_2$, with $tol_1$ and $tol_2$ fixed tolerances and $1 \le \hat{j} \le n-1$.

The sequence $\{\tilde{c}_i\}_{i=1}^{n-1}$, generated by one BIQR sweep, is thus computed until $\tilde{c}_{\tilde{j}} < tol_1$ and $|\tilde{t}^{(\tilde{j})}_{\tilde{j}-1,\tilde{j}}| + |\tilde{t}^{(\tilde{j})}_{\tilde{j},\tilde{j}+1}| < tol_2$.

Hence, the sought index $j$ is computed as the index $\bar{j}$ such that

$$\hat{c}_{\bar{j}} + \tilde{c}_{\bar{j}} \ge \hat{c}_i + \tilde{c}_i, \quad i, \bar{j} \in [\tilde{j}, \hat{j}], i \neq \bar{j},$$

i.e., the index $\bar{j}$ is chosen so that the columns of $T_{:,1:\bar{j}}$ and $T_{:,\bar{j}:n}$ are strongly linear independent.

The last column of $\hat{Q}$ in (2) depends on all the Givens coefficients $\hat{c}_i$ and $\hat{s}_i$, $i = 1,\ldots,n-1$, while the first row of $\tilde{Q}$ in (3) depends on all the Givens coefficients $\tilde{c}_i$ and $\tilde{s}_i$, $i = 1,\ldots,n-1$.

Therefore, at the first sight one can say that both the last column of $\hat{Q}$ and the first row of $\tilde{Q}$ must be computed in order to construct the sought eigenvector even though the "splitting" index $j$ is already determined.

In the sequel we show that the sought approximation of the eigenvector can be computed relying only on the knowledge of $\hat{c}_i$ and $\hat{s}_i$, $i = 1,\ldots,j-1$, and $\tilde{c}_i$ and $\tilde{s}_i$, $i = 1,\ldots,n-j+1$. In fact, once the index $j$ is determined, we observe that the "good" part of the vector (2) can be written as

$$\hat{\mathbf{x}}_{1:j} = \begin{bmatrix} -1^{n+1}\prod_{i=1}^{n-1}\hat{s}_i \\ -1^{n}\hat{c}_1\prod_{i=2}^{n-1}\hat{s}_i \\ -1^{n-1}\hat{c}_2\prod_{i=3}^{n-1}\hat{s}_i \\ \vdots \\ -1^{j+1}\hat{c}_{j-2}\prod_{i=j-1}^{n-1}\hat{s}_i \\ -1^{j}\hat{c}_{j-1}\prod_{i=j}^{n-1}\hat{s}_i \\ -1^{j-1}\hat{c}_{j}\prod_{i=j+1}^{n-1}\hat{s}_i \end{bmatrix} = \gamma^{(u)}\hat{\mathbf{x}}^{(u)},$$

while the "good" part of the vector (2) can be written as

$$\tilde{\mathbf{x}}_{n-j:n} = \begin{bmatrix} -1^{n-j-1}\tilde{c}_{n-j}\prod_{i=n-j+1}^{n-1}\tilde{s}_i \\ -1^{n-j}\tilde{c}_{n-j-1}\prod_{i=n-j}^{n-1}\tilde{s}_i \\ -1^{n-j+1}\tilde{c}_{n-j-2}\prod_{i=n-j-1}^{n-1}\tilde{s}_i \\ \vdots \\ -1^{n-1}\tilde{c}_2\prod_{i=3}^{n-1}\tilde{s}_i \\ -1^{n}\tilde{c}_1\prod_{i=2}^{n-1}\tilde{s}_i \\ -1^{n+1}\prod_{i=1}^{n-1}\tilde{s}_i \end{bmatrix} = \gamma^{(b)}\tilde{\mathbf{x}}^{(b)},$$

where $\gamma^{(u)} = \prod_{i=j+1}^{n-1} \hat{s}_i$, $\gamma^{(b)} = \prod_{i=n-j+1}^{n-1} \tilde{s}_i$,

$$\hat{\mathbf{x}}^{(u)} = \begin{bmatrix} -1^{n+1} \prod_{i=1}^{j} \hat{s}_i \\ -1^{n} \hat{c}_1 \prod_{i=2}^{j} \hat{s}_i \\ -1^{n-1} \hat{c}_2 \prod_{i=3}^{j} \hat{s}_i \\ \vdots \\ -1^{j+1} \hat{c}_{j-2} \prod_{i=j-1}^{j} \hat{s}_i \\ -1^{j} \hat{c}_{j-1} \hat{s}_j \\ -1^{j-1} \hat{c}_j \end{bmatrix}, \quad \tilde{\mathbf{x}}^{(b)} = \begin{bmatrix} -1^{n-j-1} \tilde{c}_{n-j} \\ -1^{n-j} \tilde{c}_{n-j-1} \tilde{s}_{n-j} \\ -1^{n-j+1} \tilde{c}_{n-j-2} \prod_{i=n-j-1}^{n-j} \tilde{s}_i \\ \vdots \\ -1^{n-1} \tilde{c}_2 \prod_{i=3}^{n-j} \tilde{s}_i \\ -1^{n} \tilde{c}_1 \prod_{i=2}^{n-j} \tilde{s}_i \\ -1^{n+1} \prod_{i=1}^{n-j} \tilde{s}_i \end{bmatrix}.$$

Hence, we first normalize both vectors in this way,

$$\check{\mathbf{x}}_{1:j} = \frac{\hat{\mathbf{x}}_{1:j}^{(u)}}{\hat{\mathbf{x}}_{j}^{(u)}}, \quad \check{\mathbf{x}}_{j+1:n} = \frac{\tilde{\mathbf{x}}_{2:n-j+1}^{(b)}}{\tilde{\mathbf{x}}_{1}^{(b)}},$$

i.e., we divide the first vector by its last component and the second one by its first one in order to have 1 as the $j$-th entry of the first vector and as the first entry of the second one, and finally we normalize $\check{\mathbf{x}}$ such that $\|\check{\mathbf{x}}\|_2 = 1$.

The corresponding `MATLAB` code to compute the eigenvector associated to a given eigenvalue of a symmetric tridiagonal matrix is freely available and can be downloaded at users.ba.cnr.it/iac/irmanm21/TRID_SYM.

## 5 Deflation

Once the eigenvector $\check{\mathbf{x}}$ has been computed, we can apply to it a sequence of $n-1$ Givens rotations $G_i$ in order to transform it either to $\pm\mathbf{e}_1^{(n)}$ or to $\pm\mathbf{e}_n^{(n)}$, where $\pm\mathbf{e}_i^{(n)}$, $i = 1, \ldots, n$, is the $i$th vector of the canonical basis of $\mathbb{R}^n$.

The same Givens rotations $G_i$ are applied to the matrix $T$ obtaining

$$\check{T} = G_{n-1} G_{n-2} \cdots G_1 T G_1^T \cdots G_{n-2}^T G_{n-1}^T. \tag{4}$$

If the eigenvector $\check{\mathbf{x}}$ is computed in an accurate way and satisfies particular properties, it has been shown in [9, 10] that $\check{T}$ in (4) is still tridiagonal with the entry (2, 1) equal to zero if the Givens rotations are applied in a backward fashion or the entry $(n, n-1)$ is equal to zero if the Givens rotations are applied in a forward manner. In the first case the last row and column can be dropped and the other eigenvalues to be computed are the eigenvalues of $\check{T}_{1:n-1}$. In the other case, the first row and column are removed and the other eigenvalues are the eigenvalues of $\check{T}_{2:n}$.

## 6 Numerical Examples

All the numerical experiments of this section are performed in `MATLAB Ver. 2014b`, with machine precision $\varepsilon \sim 2.22 \times 10^{-16}$. We have compared the results obtained computing the eigenvector matrix with the following techniques: `eig` of `MATLAB`, `MR`$^3$ and the proposed method, denoted by `MTV`.[3] For the second and the third method, the eigenvalues are computed by the `bisection` method [14].

For all the experiments, the tolerances $tol_1$ and $tol_2$ were chosen equal to $n\sqrt{\varepsilon}$, with $n$ the order of the matrix.

*Example 2* In this example we consider symmetric tridiagonal matrices $T_n \in \mathbb{R}^{n \times n}$, $n = 128, 256, 512, 1024$ whose elements are generated by the `MATLAB` function `randn`.

The latter matrices can be downloaded at users.ba.cnr.it/iac/irmanm21/TRID$_$SYM.

In Table 1 the orthogonality of the computed eigenvectors with the considered three methods are displayed. In column 5, the average lengths of the computed intervals in which to search the index $j$ are reported.

In Table 2 the accuracy of the residuals of the computed eigenvectors with the considered three methods are displayed.

We can conclude that the eigenvectors obtained with the proposed procedure are computed in an accurate way.

*Example 3* In this example $T_n \in \mathbb{R}^{n \times n}$, $n = 128, 256, 512, 1024$ are the Jacobi matrices associated to the Chebyshev polynomials of first kind [3], whose eigenvalues are

$$\cos\left(\frac{i\pi}{n+1}\right), \quad i = 1\ldots, n.$$

In Table 3 the orthogonality of the computed eigenvectors with the considered three methods are displayed. We do not report the average lengths of the computed intervals in which to search the index $j$ in this case, since there is no premature deflation for such matrices.

In Table 4 the accuracy of the residuals of the computed eigenvectors with the considered three methods are displayed.

We can conclude that the eigenvectors obtained with the proposed procedure are computed in an accurate way.

[3]We have used a `MATLAB` implementation of the `MR`$^3$ algorithm written by Petschow [13].

**Table 1** Accuracy of the orthogonality of the computed eigenvectors computed by `eig` of `MATLAB` (second column), by `MR`$^3$ (third column) and by the proposed method (fourth column)

| $\max_i \frac{\|X^T\mathbf{x}_i - \mathbf{e}_i^{(n)}\|_2}{n\varepsilon}$ | | | | |
|---|---|---|---|---|
| $n$ | `eig` | `MR`$^3$ | `MTV` | $\frac{\sum_{i=1}^{n}(\tilde{j}-\hat{j}+1)}{n}$ |
| 128 | $1.14 \times 10^{-1}$ | $2.01 \times 10^{1}$ | $7.05 \times 10^{1}$ | 26 |
| 256 | $5.72 \times 10^{-2}$ | $1.00 \times 10^{1}$ | $3.52 \times 10^{1}$ | 24 |
| 512 | $2.86 \times 10^{-2}$ | $5.02 \times 10^{0}$ | $1.97 \times 10^{1}$ | 24 |
| 1024 | $1.43 \times 10^{-2}$ | $3.40 \times 10^{0}$ | $3.83 \times 10^{1}$ | 24 |

Average lengths of the computed intervals in which the index $j$ is sought (fifth column)

**Table 2** Accuracy of the residuals of the eigenvectors computed by `eig` of `MATLAB` (second column), by `MR`$^3$ (third column) and by the proposed method (fourth column)

| $\max_i \frac{\|T\bar{x}_i - \lambda_i\bar{x}_i\|_2}{n\varepsilon\|T\|_2}$ | | | |
|---|---|---|---|
| $n$ | `eig` | `MR`$^3$ | `MTV` |
| 128 | $4.96 \times 10^{-2}$ | $5.33 \times 10^{0}$ | $1.87 \times 10^{1}$ |
| 256 | $1.73 \times 10^{-2}$ | $1.00 \times 10^{1}$ | $3.52 \times 10^{1}$ |
| 512 | $1.06 \times 10^{-2}$ | $5.02 \times 10^{0}$ | $1.76 \times 10^{1}$ |
| 1024 | $6.72 \times 10^{-3}$ | $5.94 \times 10^{-1}$ | $5.96 \times 10^{0}$ |

**Table 3** Accuracy of the orthogonality of the computed eigenvectors computed by `eig` of `MATLAB` (second column), by `MR`$^3$ (third column) and by the proposed method (fourth column)

| $\max_i \frac{\|X^T\mathbf{x}_i - \mathbf{e}_i^{(n)}\|_2}{n\varepsilon}$ | | | |
|---|---|---|---|
| $n$ | `eig` | `MR`$^3$ | `MTV` |
| 128 | $2.36 \times 10^{-1}$ | $2.901 \times 10^{0}$ | $2.16 \times 10^{2}$ |
| 256 | $1.18 \times 10^{-1}$ | $2.69 \times 10^{0}$ | $6.35 \times 10^{2}$ |
| 512 | $5.92 \times 10^{-2}$ | $7.26 \times 10^{1}$ | $1.03 \times 10^{1}$ |
| 1024 | $1.43 \times 10^{-2}$ | $2.99 \times 10^{0}$ | $3.83 \times 10^{1}$ |

**Table 4** Accuracy of the residuals of the eigenvectors computed by `eig` of `MATLAB` (second column), by `MR`$^3$ (third column) and by the proposed method (fourth column)

| $\max_i \frac{\|T\bar{x}_i - \lambda_i\bar{x}_i\|_2}{n\varepsilon\|T\|_2}$ | | | |
|---|---|---|---|
| $n$ | `eig` | `MR`$^3$ | `MTV` |
| 128 | $1.67 \times 10^{-1}$ | $2.05 \times 10^{0}$ | $1.52 \times 10^{2}$ |
| 256 | $8.37 \times 10^{-2}$ | $1.45 \times 10^{1}$ | $1.08 \times 10^{1}$ |
| 512 | $4.18 \times 10^{-2}$ | $7.26 \times 10^{0}$ | $1.05 \times 10^{2}$ |
| 1024 | $6.72 \times 10^{-3}$ | $9.48 \times 10^{-1}$ | $5.21 \times 10^{0}$ |

## 7 Conclusions

Recently, Malyshev and Dhillon have proposed an algorithm for deflating the tridiagonal matrix, once an eigenvalue has been computed. Starting from the above mentioned algorithm, a method for computing the eigenvectors of a symmetric tridiagonal matrix $T$ has been proposed. It requires the computation of an index $j$ which determines the premature deflation in the implicit QR method. The index $j$ is computed considering the two sequences of cosines generated by a sweep of forward and backward QR method with shift the computed eigenvalue. The sought eigenvector is obtained form the first $j$ Givens coefficients generated by the forward implicit QR method and from the last $n - j$ Givens coefficients generated by the backward implicit QR method.

The overall complexity for computing an eigenvector depends linearly on the size of the matrix.

The numerical tests show the reliability of the proposed technique.

**Acknowledgements** The authors wish to thank the anonymous reviewers for their constructive remarks that helped improving the proposed algorithm and the presentation of the results.

The authors would like to thank Paolo Bientinesi and Matthias Petschow for providing their MATLAB implementation of the MR3 algorithm, written by Matthias Petschow.

The work of the author "Nicola Mastronardi" is partly supported by GNCS–INdAM and by CNR under the Short Term Mobility Program. The work of the author "Harold Taeter" is supported by INdAM-DP-COFUND-2015, grant number: 713485.

## References

1. Anderson, E., Bai, Z., Bischof, C., Blackford, L., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' Guide, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia (1999)
2. Cuppen, J.J.M.: A divide and conquer method for the symmetric tridiagonal eigenproblem. Numer. Math. **36**, 177–195 (1981)
3. Davis, P., Rabinowitz, P.: Methods of Numerical Integration, 2nd edn. Academic Press, Cambridge (1984)
4. Dhillon, I., Malyshev, A.: Inner deflation for symmetric tridiagonal matrices. Linear Algebra Appl. **358**, 139–144 (2003)
5. Dhillon, I., Parlett, B.: Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices. Linear Algebra Appl. **387**, 1–28 (2004)
6. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2013)
7. Gu, M., Eisenstat, S.: A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem. SIAM J. Matrix Anal. Appl. **16**(1), 172–191 (1995)
8. Horn, R., Johnson, C.: Topics in Matrix Analysis. Cambridge University Press, New York (1991)
9. Mastronardi, N., Van Dooren, P.: Computing the Jordan structure of an eigenvalue. SIAM J. Matrix Anal. Appl. **38**, 949–966 (2017)
10. Mastronardi, N., Van Dooren, P.: The $QR$–steps with perfect shifts. SIAM J. Matrix Anal. Appl. **39**, 1591–1615 (2018)

11. Parlett, B.N.: The Symmetric Eigenvalue Problem. Society for Industrial and Applied Mathematics, Philadelplhia (1997)
12. Parlett, B.N., Le, J.: Forward instability of tridiagonal QR. SIAM J. Matrix Anal. Appl. **14**(1), 279–316 (1993)
13. Petschow, M., Quintana-Ortí, E., Bientinesi, P.: Improved accuracy and parallelism for MRRR–based eigensolvers–a mixed precision approach. SIAM J. Sci. Comput. **36**(2), C240–C263 (2014)
14. Wilkinson, J., Bauer, F., Reinsch, C.: Linear Algebra. Handbook for Automatic Computation. Springer, Berlin (2013)
15. Willems, P.R., Lang, B.: Twisted factorizations and qd–type transformations for the MR3 algorithm–new representations and analysis. SIAM J. Matrix Anal. Appl. **33**(2), 523–553 (2012)
16. Willems, P.R., Lang, B.: A framework for the MR3 algorithm: theory and implementation. SIAM J. Sci. Stat. Comput. **35**(2), A740–A766 (2013)