Noname manuscript No.
(will be inserted by the editor)

Computing the eigenvectors of nonsymmetric tridiagonal matrices

Teresa Laudadio · Nicola Mastronardi · Paul Van Dooren

Received: date / Accepted: date

Abstract The computation of the eigenvalue decomposition of matrices is one of the most investigated problems in numerical linear algebra. In particular, real nonsymmetric tridiagonal eigenvalue problems arise in a variety of applications. In this paper the problem of computing an eigenvector corresponding to a known eigenvalue of a real nonsymmetric tridiagonal matrix is considered, developing an algorithm that combines part of a QR sweep and part of a QL sweep, both with the shift equal to the known eigenvalue. The numerical tests show the reliability of the proposed method.

Keywords Nonsymmetric tridiagonal matrices · eigenvectors · Bessel matrices

Mathematics Subject Classification (2010) 15A18 · 65F15

1 Introduction

Real nonsymmetric tridiagonal eigenvalue problems arise in a variety of applications. For example, the nonsymmetric eigenvalue problem can be reduced to nonsymmetric tridiagonal form in a finite number of steps [17,5,18]. In the sparse case, the nonsymmetric Lanczos algorithm produces a nonsymmetric tridiagonal matrix by the biorthogonal Lanczos process [9]. The zeros of generalized Bessel polynomials can be computed as the eigenvalues of the associated Bessel matrices, which are nonsymmetric tridiagonal [6,15]. Other eigenvalue problems involving nonsymmetric tridiagonal matrices can be found in [1].

The work of T.L. and N.M. was partly supported by GNCS–INdAM. The work of P.V.D. was partly supported by CNR, Italy under the Short Term Mobility Program.

Istituto per le Applicazioni del Calcolo, CNR, via Amendola 122/D, 70126 Bari, Italy.
Department of Mathematical Engineering, Catholic University of Louvain, Louvain-la-Neuve, Belgium.
E-mail: {t.laudadio, n.mastronardi}@cnr.it, paul.vandooren@uclouvain.be

Some methods to compute the eigenvalues of nonsymmetric tridiagonal matrices are available in the literature. The one described in [1] requires the evaluation of $p(\lambda)/p'(\lambda) = -1/\text{trace}\left((T-\lambda I)^{-1}\right)$, with $p(\lambda)$ the characteristic polynomial of the matrix T, which is done by exploiting the QR factorization of $T-\lambda I$ and the semiseparable structure of $(T-\lambda I)^{-1}$. The algorithm described in [4] relies on the LR method [16]. The implicit QR method [7,8] can be considered for computing the eigenstructure of a tridiagonal matrix. Unfortunately, this method does not exploit the tridiagonal structure of the problem and, therefore, has a complexity depending cubically on the size of the matrix. In this manuscript, we focus on the computation of the eigenvectors of tridiagonal matrices once the corresponding eigenvalues are known.

It is well known that, in exact arithmetic, after one sweep of the QR (QL) method with shift equal to an eigenvalue, this eigenvalue is displayed in the bottom–right (top–left) of the matrix and can then be deflated [10]. Unfortunately, both methods can suffer from forward instability when working in finite precision arithmetic [14,19]. In order to avoid this instability phenomenon, we combine one sweep of both the QR and QL methods with shift equal to an eigenvalue λ.

Here we focus on computing the left eigenvector associated to a known eigenvalue of a nonsymmetric tridiagonal matrix. For the sake of brevity, we omit to describe the algorithm for computing the right eigenvector since this problem is completely analogous. The numerical examples show the reliability of the proposed approach.

The manuscript is organized as follows. In Section 2 the notations used and some basic definitions are given. In Section 3 the main features of the QR method are outlined. The proposed algorithm is described in Section 4. In Section 5 it is shown how to avoid complex arithmetic for the case of complex conjugate eigenvalues. In Section 6 we give a number of numerical examples and we end with a section of concluding remarks.

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 $1 \le i \le i+k \le n$, and columns $j, j+1, j+2, \ldots, j+\ell$, with $1 \le j \le j+\ell \le n$, is denoted by $B_{i:i+k,j:j+\ell}$.
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 ε_M.
The ith 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 The columns of $B \in \mathbb{R}^{m\times n},\ m \geq n$, are said ε–linear dependent if $\sigma_n(B) \leq \varepsilon\|B\|_2$, with $\sigma_n(B)$ the n–th singular value of B, and ε an arbitrary tolerance.

3 Computing an eigenvector with the QR and the QL method

Let $T \in \mathbb{R}^{n\times n}$ be an irreducible nonsymmetric tridiagonal matrix

$$T = \begin{bmatrix} \alpha_1 & \gamma_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix},$$

i.e., $\beta_i \neq 0,\ \ \gamma_i \neq 0,\ i = 1,\ldots,n-1$, and let $\lambda_i,\ i = 1,\ldots,n$, be its eigenvalues. Since T is irreducible, the geometric multiplicity of each eigenvalue is 1.

We first describe how the left eigenvector $\mathbf{y}$ associated to an eigenvalue λ of T can be obtained applying one sweep of the QR or the QL method with shift λ. Unfortunately, both methods can suffer from forward instability when working in finite precision arithmetic. Here we analyze the reasons of such a behaviour and then propose an algorithm, based on a suitable combination of the aforementioned methods, to overcome the instability phenomenon.

3.1 QR method

Let $\kappa \in \mathbb{C}$. Let $\hat{Q}\hat{R}$ be the QR factorization of $T(\kappa)$, with

$$\hat{R} = \begin{bmatrix} \hat{\rho}_1 & \hat{\gamma}_1 & \hat{\beta}_1 & & \\ & \hat{\rho}_2 & \hat{\gamma}_2 & \ddots & \\ & & \ddots & \ddots & \hat{\beta}_{n-2} \\ & & & \hat{\rho}_{n-1} & \hat{\gamma}_{n-1} \\ & & & & \hat{\rho}_n \end{bmatrix}$$

and $\hat{Q} = \hat{G}_{n-1}^H \hat{G}_{n-2}^H \cdots \hat{G}_1^H$, where $\hat{G}_i$ are the Givens rotations[1]

$$\hat{G}_i = \begin{bmatrix} I_{i-1} & & & \\ & c_i & s_i & \\ & -\bar{s}_i & c_i & \\ & & & I_{n-i-1} \end{bmatrix}, \quad c_i^2 + |s_i|^2 = 1, \quad i = 1,\ldots,n-1.$$

[1] The coefficients $c_i,\ i = 1,\ldots,n-1$, of the Givens rotations are real [10, p. 243].

Hence, $\hat{Q}$ is the unitary upper Hessenberg matrix

$$\hat{Q} = \begin{bmatrix} c_1 & -s_1c_2 & s_1s_2c_3 & \ddots & (-1)^{n-2}c_{n-1}\prod_{i=1}^{n-2}s_i & (-1)^{n-1}\prod_{i=1}^{n-1}s_i \\ \bar{s}_1 & c_1c_2 & -c_1s_2c_3 & \ddots & (-1)^{n-3}c_1c_{n-1}\prod_{i=2}^{n-2}s_i & (-1)^{n-2}c_1\prod_{i=2}^{n-1}s_i \\ & \bar{s}_2 & c_2c_3 & \ddots & \vdots & \vdots \\ & & \ddots & \ddots & -c_{n-3}s_{n-2}c_{n-1} & c_{n-3}s_{n-2}s_{n-1} \\ & & & \bar{s}_{n-2} & c_{n-2}c_{n-1} & -c_{n-2}s_{n-1} \\ & & & & \bar{s}_{n-1} & c_{n-1} \end{bmatrix}.$$

In exact arithmetic, if $\kappa \equiv \lambda_i,\ i \in \{1,\dots,n\}$, it turns out that $\hat{\rho}_n = 0$ and

$$\mathbf{y}_i \equiv \hat{Q}(:,n) \tag{1}$$

is the left eigenvector associated to κ.

As mentioned before, in finite precision arithmetic forward instability can occur in the QR factorization of $T(\lambda_i)$, and the last column of $\hat{Q}$ may be far from the sought eigenvector [13,14].

In particular, defining $\hat{R}^{(0)}(\lambda_i) = T - \lambda_i I_n$, $\hat{R}^{(i)}(\lambda_i) = \hat{G}_i\hat{R}^{(i-1)}(\kappa),\ i = 1,\dots,n-1$, forward instability occurs at step j of the QR factorization if and only if $\hat{R}^{(j)}_{1:j+1,1:j+1}(\kappa)$ is nearly singular and c_j is almost negligible. Such a phenomenon was already described in [14] for the symmetric case.

3.2 QL method

Let $\tilde{Q}\tilde{L} = T - \kappa I_n$ be the QL factorization of $T(\kappa)$, obtained applying the sequence of Givens rotations

$$\tilde{G}_i = \begin{bmatrix} I_{n-i-1} & & & \\ & g_i & h_i & \\ & -\bar{h}_i & g_i & \\ & & & I_{i-1} \end{bmatrix}, \quad g_i^2 + |h_i|^2 = 1, \quad i = 1,\dots,n-1.$$

It turns out that

$$\tilde{L} = \begin{bmatrix} \tilde{\rho}_1 & & & \\ \tilde{\gamma}_1 & \tilde{\rho}_2 & & \\ \tilde{\beta}_1 & \tilde{\gamma}_2 & \ddots & \\ & \ddots & \ddots & \tilde{\rho}_{n-1} \\ & & \tilde{\beta}_{n-2} & \tilde{\gamma}_{n-1}\ \tilde{\rho}_n \end{bmatrix}$$

and $\tilde{Q} = \tilde{G}_{n-1}^H \tilde{G}_{n-2}^H \cdots \tilde{G}_1^H$ is the lower Hessenberg matrix

$$\tilde{Q} = \begin{bmatrix} g_{n-1} & \bar{h}_{n-1} & & & & \\ -g_{n-2}h_{n-1} & g_{n-2}g_{n-1} & \bar{h}_{n-2} & & & \\ g_{n-3}h_{n-2}h_{n-1} & -g_{n-3}h_{n-2}g_{n-1} & \ddots & \ddots & & \\ \vdots & \vdots & \ddots & g_3g_2 & \bar{h}_2 & \\ (-1)^{n-2}g_1\prod_{i=2}^{n-1}h_i & (-1)^{n-3}g_1g_{n-1}\prod_{i=2}^{n-2}h_i & \ddots & -g_3h_2g_1 & g_2g_1 & \bar{h}_1 \\ (-1)^{n-1}\prod_{i=1}^{n-1}h_i & (-1)^{n-2}g_{n-1}\prod_{i=1}^{n-2}h_i & \ddots & g_3h_2h_1 & -g_2h_1 & g_1 \end{bmatrix}.$$

In exact arithmetic, if $\kappa \equiv \lambda_i,\ i \in \{1,\ldots,n\}$, then $\tilde{\rho}_1 = 0$ and

$$\mathbf{y}_i \equiv \tilde{Q}(:,1) \tag{2}$$

is the left eigenvector associated to λ_i.

Also in this case, forward instability can occur, and the first column of $\tilde{Q}$ may be different from the sought eigenvector.

In particular, defining $\tilde{L}^{(0)}(\lambda_i) = T(\lambda_i),\ \tilde{L}^{(i)}(\lambda_i) = \tilde{G}_i\tilde{L}^{(i-1)}(\lambda_i), i = 1,\ldots,n-1$, forward instability occurs at step j of the QL factorization if and only if $\tilde{L}^{(j)}_{n-j:n,n-j:n}(\kappa)$ is very close to be singular and the first entry of the eigenvector corresponding to its smallest eigenvalue is tiny [14]. By (1), this entry is given by g_j. Therefore, forward instability happens if $\tilde{L}^{(j)}_{n-j:n,n-j:n}(\lambda_i)$ is nearly singular and g_j is negligible.

To examine in which step of the QR and QL methods forward instability can occur, let us consider the following Corollary [11, p. 149].

Corollary 1 *Given $A \in \mathbb{C}^{m\times n}$, let B be a submatrix obtained by deleting r columns from A. Then*

$$\sigma_k(A) \geq \sigma_k(B) \geq \sigma_{k+r}(A),\ \ k = 1,\ldots,\min\{m,n\}, \tag{3}$$

where $\sigma_\ell(A) \equiv 0$, if $\ell > \min\{m,n\}$.

Let λ be an eigenvalue of T. Let us denote the singular values of $\hat{R}^{(i)}_{1:i+1,:}(\lambda)$ by $\sigma_j(\hat{R}^{(i)}_{1:i+1,:}(\lambda)),\ j = 1,\ldots,i+1$, and the singular values of $\tilde{L}^{(i)}_{n-i:n,:}(\lambda)$ by $\sigma_j(\tilde{L}^{(i)}_{n-i:n,:}(\lambda)),\ j = 1,\ldots,i+1$, for $i = 1,\ldots,n-1$, respectively. By Corollary 1, we have

$$\begin{cases} \sigma_{i+1}(\hat{R}^{(i)}_{1:i+1,:}(\lambda)) \geq \sigma_{i+2}(\hat{R}^{(i+1)}_{1:i+2,:}(\lambda)), \\ \sigma_{i+1}(\tilde{L}^{(i)}_{n-i:n,:}(\lambda)) \geq \sigma_{i+2}(\tilde{L}^{(i+1)}_{n-i-1:n,:}(\lambda)), \end{cases} \quad i = 1,\ldots,n-2.$$

Suppose that

$$\sigma_{n-1}(T(\lambda)) \gg \varepsilon > \sigma_n(T(\lambda)) = 0.$$

The following lemma shows that the matrices $\hat{R}^{(j)}_{1:j+1,:}(\lambda)$ and $\tilde{L}^{(n-j-2)}_{j+2:n,:}$ can not have both a singular value below ε.

Lemma 1 *Let $k_1, k_2 \in \{1, \ldots, n\}$, $k_1 \neq k_2$, and*

$$\Delta = T(\lambda) \left[\mathbf{e}_{k_1} \middle| \mathbf{e}_{k_2} \right].$$

Then

$$\|\Delta\|_F \geq \|\Delta\|_2 \geq \sigma_{n-1}\left(T(\lambda)\right), \tag{4}$$

where $\sigma_{n-1}\left(T(\lambda)\right)$ is the $(n-1)$–th singular value of $T(\lambda)$.

Proof By (3), we have that

$$\sigma_1(\Delta) = \|\Delta\|_2 \geq \sigma_{n-1}\left(T(\lambda)\right),$$

with $\sigma_1(\Delta)$ the largest singular value of Δ, from which (4) follows. □

Let us consider, for every j, the factorization of $T(\lambda)$,

$$T(\lambda) = \left[\begin{array}{c|c} \hat{Q}^{(j)} & \\ \hline & \tilde{Q}^{(n-j-2)} \end{array}\right] \left[\begin{array}{c} \hat{R}^{(j)}_{1:j+1,:}(\lambda) \\ \hline \tilde{L}^{(n-j-2)}_{j+2:n,:}(\lambda) \end{array}\right],$$

with $\hat{Q}^{(j)} = \hat{G}_j^H \hat{G}_{j-1}^H \cdots \hat{G}_1^H$, . and $\tilde{Q}^{(n-j-2)} = \tilde{G}_{n-j-2}^H \tilde{G}_{n-j-3}^H \cdots \tilde{G}_2^H \tilde{G}_1^H$. It follows from Lemma 1 that only one of the 2 submatrices can be ε–dependent, for $\varepsilon \leq \sigma_{n-1}\left(T(\lambda)\right)$.

Let us define $\hat{\sigma}_i = \sigma_i(\hat{R}^{(i-1)}_{1:i,:}(\lambda))$ and $\tilde{\sigma}_i = \sigma_{n-i+1}(L^{(n-i)}_{i:n,:}(\lambda))$, $i = 1, \ldots, n$.

Lemma 2 *The sequence*

$$\{\max(\hat{\sigma}_i, \tilde{\sigma}_i)\}_{i=1}^n = \{\hat{\sigma}_1, \ldots, \hat{\sigma}_{j-1}, \max(\hat{\sigma}_j, \tilde{\sigma}_j), \tilde{\sigma}_{j+1}, \ldots, \tilde{\sigma}_n)\}$$

is unimodal and has its minimum value equal to $\max(\hat{\sigma}_j, \tilde{\sigma}_j)$ *for some* $j \in \{1, \ldots, n\}$. *Moreover, this is the only element of the sequence that can be smaller that $\sigma_{n-1}\left(T(\lambda)\right)$.*

Hence, it follows from the perturbation analysis of the QR factorization [2] that forward instability in the calculation of $\hat{Q}$, and therefore of the cosines c_i, is not encountered in the first $j-1$ steps of the QR factorization and forward instability in the calculation of $\tilde{Q}$ is not encountered in the first $n-j-1$ steps of the QL factorization and therefore of the cosines g_i. In order to detect the relevant index j we can make use of the sequences $\{c_j\}_{j=1}^{n-1}$ and $\{g_j\}_{j=1}^{n-1}$, as shown in the following examples.

Example 1 Let $T \in \mathbb{R}^{n \times n}$, $n = 100$, be a nonsymmetric irreducible tridiagonal matrix whose entries are generated by the `matlab` function `randn` and let λ be an eigenvalue of T. Let $(\lambda, \mathbf{y})$ be a left eigenpair of T, obtained applying few steps of inverse iteration to the corresponding couple yielded by the function `eig` of `matlab`. We remark that, for such matrices, the behaviour we are going to describe is observed for all the eigenvalues.

The sequence of cosines $\{c_i\}_{i=1}^{n-1}$ generated by the QR method with shift equal to λ (denoted by "$*$"), the sequence $\{\hat{\sigma}_i\}_{i=1}^n$, (denoted by "$\nabla$"), the modulus of the entries of the left eigenvector $\mathbf{y}$ associated to λ (denoted by

"$\diamond$") and the modulus of the entries of the last column of $\hat{Q}$ (denoted by "$+$") are reported in Figure 1 on a logarithmic scale.

The sequence of cosines $\{g_i\}_{i=1}^{n-1}$ generated by the QL method with shift equal to λ (denoted by "$*$"), the sequence $\{\tilde{\sigma}_i\}_{i=1}^{n}$, (denoted by "$\nabla$"), the modulus of the entries of the left eigenvector $\mathbf{y}$ associated to λ (denoted by "$\diamond$") and the modulus of the entries of the first column of $\tilde{Q}$ (denoted by "$+$") are reported in Figure 2 on a logarithmic scale. In Figure 1 we can observe that

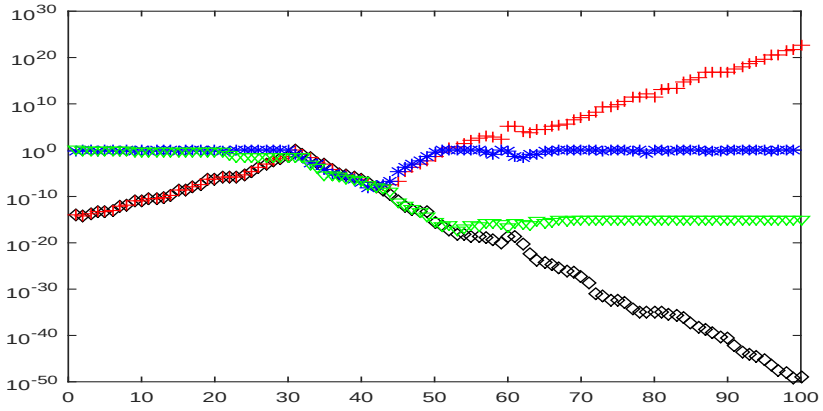

Fig. 1 Example 1 : plot of $\{c_i\}_{i=1}^{n-1}$ ("$*$"), $\{\hat{\sigma}_i\}_{i=1}^{n}$, ("$\nabla$"), modulus of the entries of the left eigenvector $\mathbf{y}$ associated to λ ("$\diamond$") and modulus of the entries of the last column of $\hat{Q}$ ("$+$") computed by one sweep of the QR method applied to $T(\lambda)$.

the elements of sequence $\{c_i\}_{i=1}^{n-1}$ have a behaviour similar to the elements of the sequence $\{\hat{\sigma}_i\}_{i=1}^{n}$ until their value is above $\sqrt{\varepsilon_M}$. Then forward instability occurs and the rest of the elements of the first sequence depart from those of the second one. Analogously, the moduli of the entries of the last column of $\hat{Q}$ match those of $\mathbf{y}$ until forward instability occurs.

On the other hand, Figure 2 shows that the last entries of the sequences $\{g_i\}_{i=1}^{n-1}$ and $\{\tilde{\sigma}_i\}_{i=1}^{n}$, have a similar behaviour until the entries of the first sequence are above $\sqrt{\varepsilon_M}$. This phenomenon was also described in [14] for symmetric tridiagonal matrices. The entries of $\mathbf{y}$ and the first column of $\tilde{Q}$ have a similar behaviour, too. The component–wise errors of $|\mathbf{y} - \hat{Q}(:,n)|$ (denoted by "$\times$") and of $|\mathbf{y} - \tilde{Q}(:,1)|$ (denoted by "$+$"), the sequences $\{\hat{\sigma}_i\}_{i=1}^{n}$ (denoted by "$\diamond$") and $\{\tilde{\sigma}_i\}_{i=1}^{n}$ (denoted by "$\circ$") and $\{c_i\}_{i=1}^{n-1}$ (denoted "$\circ$"), $\{g_i\}_{i=1}^{n-1}$("$*$"), the machine precision ε_M (denoted by a blue line) are depicted on a logarithmic scale in Figure 3.

We can observe that the component–wise errors of $|\mathbf{y} - \hat{Q}(:,1)|$ are always below ε_M before the index 30, while and $|\mathbf{y} - \tilde{Q}(:,1)|$ are always below ε_M after the index 30, as predicted by Lemma 1.

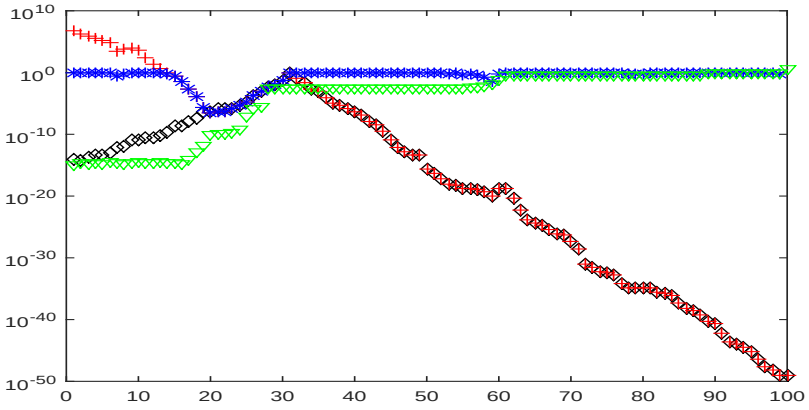

Fig. 2 Example 1 : plot of $\{g_i\}_{i=1}^{n-1}$ ("$*$"), $\{\tilde{\sigma}_i\}_{i=1}^{n}$, ("$\triangledown$"), the modulus of the entries of the left eigenvector $\mathbf{y}$ associated to λ ("$\diamond$") and the modulus of the entries of the first column of $\tilde{Q}$ ("$+$") computed by one sweep of the QL method applied to $T(\lambda)$.

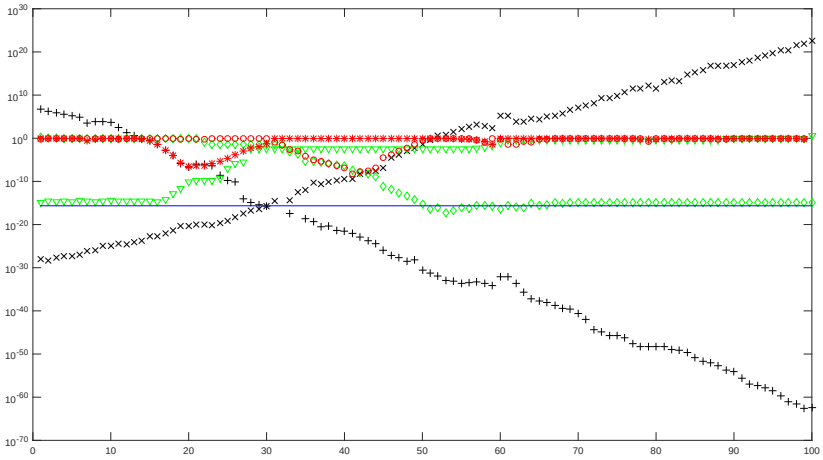

Fig. 3 Example 1 : plot of $|\mathbf{y} - \hat{Q}(:,n)|$ ("$\times$"), $|\mathbf{y} - \tilde{Q}(:,1)|$ ("$+$"), $\{\hat{\sigma}_i\}_{i=1}^{n}$ ("$\diamond$"), $\{\tilde{\sigma}_i\}_{i=1}^{n}$ ("$\triangledown$"), $\{c_i\}_{i=1}^{n-1}$ ("$\circ$") and $\{g_i\}_{i=1}^{n-1}$("$*$"), the machine precision ε_M (blue line) in logarithmic scale.

Hence, for each λ_i there is a crucial index j such that

$$\begin{cases} |\mathbf{y}_k - \hat{Q}(k,1)| \leq \upsilon \cdot \varepsilon_M, & 1 \leq k \leq j \\ |\mathbf{y}_k - \tilde{Q}(k,1)| \leq \upsilon \cdot \varepsilon_M, & j+1 \leq k \leq n, \end{cases}$$

with $\upsilon \sim O(n)$.

Summarizing, the entries of the last column of $\hat{Q}$ are accurately computed from the top until forward instability occurs in the QR sweep, i.e., as soon as there is an index $j \in \{1, \ldots, n\}$ such that $T_{1:j,:}$ has nearly dependent columns. On the other hand, forward instability in the QL sweep can only occur after $n - j - 1$ steps and the entries of first column of $\tilde{Q}$ are accurately computed from the bottom. Hence, it is crucial to determine the index j correctly.

Remark 1 We remark that, while the sequences $\{\hat{\sigma}_i\}_{i=1}^{n}$ and $\{\tilde{\sigma}_i\}_{i=1}^{n}$ are monotonic, the sequences $\{c_i\}_{i=1}^{n-1}$ and $\{g_i\}_{i=1}^{n-1}$ are not. However, we show in the next section how to determine the sought index j by looking at the entries of these sequences.

4 Combining the QR and QL methods for computing an eigenvector

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

Let $\{\hat{c}_i\}_{i=1}^{n}$ and $\{\tilde{g}_i\}_{i=1}^{n}$ be, respectively, the sequences of the coefficients of the Givens rotations needed to compute the QR and QL factorizations of $T(\lambda)$ in exact arithmetic.

If $\sigma_{n-1}(T(\lambda)) \gg \sigma_n(T(\lambda))$ and forward instability occurs at step j of the QR method with shift λ, the sequence $\{c_i\}_{i=1}^{n}$ begins to depart from the sequence $\{\hat{c}_i\}_{i=1}^{n}$ around the index j. As a consequence, the last column of $\hat{Q}$ matches $\mathbf{y}$ in the first j entries.

Analogously, the sequence $\{g_i\}_{i=1}^{n}$ begins to depart from the sequence $\{\tilde{g}_i\}_{i=1}^{n}$ around the index $n - j$ and the last $n - j$ entries of the first column of $\tilde{Q}$ match the corresponding entries of $\mathbf{y}$.

Therefore, the sought index j can be computed in the following way.

We set $i_1 = 1, \quad i_2 = n$, $R = T(\lambda), \quad L = T(\lambda)$, and compute the Givens rotations

$$\hat{G}_{i_1} = \begin{bmatrix} I_{i_1-1} & & \\ & \begin{matrix} c_{i_1} & s_{i_1} \\ -s_{i_1} & c_{i_1} \end{matrix} & \\ & & I_{n-i_1-1} \end{bmatrix} \quad \text{and} \quad \tilde{G}_{i_2-1} = \begin{bmatrix} I_{i_2-2} & & \\ & \begin{matrix} g_{i_2} & h_{i_2} \\ -h_{i_2} & g_{i_2} \end{matrix} & \\ & & I_{n-i_2} \end{bmatrix},$$

involved respectively in the corresponding step of the QR and QL methods with shift λ.
If $c_{i_1} \geq g_{i_2-1}$ then $\hat{G}_{i_1}$ is applied to $R : R \leftarrow \hat{G}_{i_1} R$ and we set $i_1 = i_1 + 1$; else $\tilde{G}_{i_2}$ is applied to $L : L \leftarrow \tilde{G}_{i_2} L$ and we set $i_2 = i_2 - 1$.
We then repeat this procedure until $i_1 > i_2$.

Observe that the last column of $\hat{Q}$ in (1) depends on all the Givens coefficients c_i and s_i, $i = 1, \ldots, n-1$, and the first column of $\tilde{Q}$ in (2) depends on all the Givens coefficients g_i and h_i, $i = 1, \ldots, n-1$.

At the first sight one can say that both the last column of $\hat{Q}$ and the first column 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 c_i and s_i, $i = 1, \ldots, j-1$, and g_i and h_i, $i = 1, \ldots, n-j+1$. In fact, once the index j is determined, we observe that the "good" part of the vector (1) can be written as

$$\hat{\mathbf{y}}_{1:j} = \begin{bmatrix} (-1)^{n-1} \prod_{i=1}^{n-1} s_i \\ (-1)^{n-2} c_1 \prod_{i=2}^{n-1} s_i \\ (-1)^{n-3} c_2 \prod_{i=3}^{n-1} s_i \\ \vdots \\ (-1)^{n-j+2} c_{j-3} \prod_{i=j-2}^{n-1} s_i \\ (-1)^{n-j+1} c_{j-2} \prod_{i=j-1}^{n-1} s_i \\ (-1)^{n-j} c_{j-1} \prod_{i=j}^{n-1} s_i \end{bmatrix} = \gamma^{(u)} \hat{\mathbf{y}}^{(u)},$$

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

$$\tilde{\mathbf{y}}_{j:n} = \begin{bmatrix} (-1)^{j-1} g_{n-j} \prod_{i=n-j+1}^{n-1} h_i \\ (-1)^{j} g_{n-j-1} \prod_{i=n-j}^{n-1} h_i \\ (-1)^{j+1} g_{n-j-2} \prod_{i=n-j-1}^{n-1} h_i \\ \vdots \\ (-1)^{n-3} g_2 \prod_{i=3}^{n-1} h_i \\ (-1)^{n-2} g_1 \prod_{i=2}^{n-1} h_i \\ (-1)^{n-1} \prod_{i=1}^{n-1} h_i \end{bmatrix} = \gamma^{(b)} \tilde{\mathbf{y}}^{(b)},$$

where $\gamma^{(u)} = (-1)^{n-j} \prod_{i=j}^{n-1} s_i$, $\gamma^{(b)} = (-1)^{j-1} \prod_{i=n-j+1}^{n-1} h_i$,

$$\hat{\mathbf{y}}^{(u)} = \begin{bmatrix} (-1)^{j-1} \prod_{i=1}^{j-1} s_i \\ (-1)^{j-2} c_1 \prod_{i=2}^{j-1} s_i \\ (-1)^{j-3} c_2 \prod_{i=3}^{j-1} s_i \\ \vdots \\ (-1)^{2} c_{j-3} \prod_{i=j-2}^{j-1} s_i \\ -c_{j-2} s_j \\ c_{j-1} \end{bmatrix}, \quad \tilde{\mathbf{y}}^{(b)} = \begin{bmatrix} g_{n-j} \\ -g_{n-j-1} h_{n-j} \\ (-1)^{2} g_{n-j-2} \prod_{i=n-j-1}^{n-j} h_i \\ \vdots \\ (-1)^{n-j-2} g_2 \prod_{i=3}^{n-j} h_i \\ (-1)^{n-j-1} g_1 \prod_{i=2}^{n-j} h_i \\ (-1)^{n-j} \prod_{i=1}^{n-j} h_i \end{bmatrix}.$$

Hence, we first normalize both vectors in this way,

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

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 $\check{\mathbf{y}}$, i.e., $\check{\mathbf{y}} = \check{\mathbf{y}}/\|\check{\mathbf{y}}\|_2$.

```
function[y] =left_eigv(T, λ, n)
R = T − λI_n;   L = R;
i_1 = 1; i_2 = n;
G_1 = givens(R(i_1, i_1), R(i_1 + 1, i_1));   G_2 = givens(L(i_2, i_2), L(i_2 − 1, i_2));
c_1(1) = G_1(i_1, 1);   s_1(1) = G_1(i_1, 2);  c_2(n − 1) = G_2(i_1, 1);  s_2(i_2 − 1) = G_2(1, 2);
while i_1 <= i_2,
      if c_1(i_1) > c_2(i_2),
         R(i_1 : i_1 + 1, :) = G_1 ∗ R(i_1 : i_1 + 1, :);
         i_1 = i_1 + 1;
         if i_1 < n,
            G_1 = givens(R(i_1, i_1), R(i_1 + 1, i_1));
            c_1(i_1) = G_1(1, 1);   s_1(i_1) = G_1(1, 2);
         end
       else
         G_2 = G_2.′;
         L(i_2 − 1 : i_2, :) = G_2 ∗ L(i_2 − 1 : i_2, :);
         i_2 = i_2 − 1;
         if i_2 > 1,
            G_2 = givens(L(i_2, i_2), L(i_2 − 1, i_2));
            c_2(i_2 − 1) = G_2(1, 1);   c_2(i_2 − 1) = G_2(1, 1);
         end
      end
end
% computation of the eigenvector
j = min(i_1, i_2);
y_1 = zeros(n, 1);   y_2 = zeros(n, 1);
y_1(j) = 1;   y_2(j) = 1;
for i = j − 1 : −1 : 1,
    G = [ c_1(i)   s_1(i) ; −s_1(i)  c_1(i) ];
    y_1(i : i + 1) = G.^T ∗ y_1(i : i + 1);
end
for i = j : n − 1,
    G = [ c_2(i)   s_2(i) ; −s_2(i)  c_2(i) ];
    y_2(i : i + 1) = G ∗ y_2(i : i + 1);
end
y_1 = y_1/y_1(j);   y_2 = y_2/y_2(j);   y = [y_1(1 : j); y_2(j + 1 : n)];
y = y/‖y‖_2;
```

Table 1 `Matlab`–like code for computing the eigenvector of a nonsymmetric tridiagonal matrix, known the corresponding eigenvalue λ.

The corresponding `matlab`–like code is listed in Table 1.

Instead of stopping the algorithm when $i_1 > i_2$, we perform j_f more steps of the QR method and j_b more steps of the QL method, with $j_f = \min\{j+k, n\}$ and $j_b = \max\{1, j-k\}$ and $k \in \{1, \ldots, n\}$. In our experiments we choose $k = \lfloor\sqrt{n}\rfloor$. Then, we compute the new vector $\check{\mathbf{c}} = [\check{c}_1, \ldots, \check{c}_n]^T$, with $\check{c}_i = c_i + g_{i+1},\ i = 1, \ldots, n$, and choose $j \in [j_b, j_f]$ such that $\check{c}_j \geq \check{c}_\ell,\ \ell \in [j_b, j_f],\ \ j \neq \ell$.

The corresponding `matlab` code can be obtained from the authors.

5 Avoiding complex arithmetic

If T is a real nonsymmetric tridiagonal matrix, its eigenvalues can be complex conjugate. The factors of the QR and QL factorizations of $T(\lambda)$ with shift a complex conjugate eigenvalue λ, are complex as well. To avoid complex arithmetic, the implicit QR/QL (IQR/IQL) method with double shift [7,8], can be considered. In exact arithmetic, after one sweep of IQR applied to T with double shift λ and $\bar{\lambda}$, with $\bar{\lambda}$ the complex conjugate of λ, has a block Hessenberg structure

$$\begin{bmatrix} H_1^{(r)} & B \\ & H_2^{(r)} \end{bmatrix},$$

with $H_1^{(r)} \in \mathbb{R}^{(n-2)\times(n-2)}$ and $H_2^{(r)} \in \mathbb{R}^{2\times 2}$ both Hessenberg, and λ and $\bar{\lambda}$ are the eigenvalues of $H_2^{(r)}$. Moreover, the last two columns of the Q factor are the real and the imaginary part of the associated eigenvectors.

Unfortunately, one sweep of the IQR method (IQL method) transforms the tridiagonal matrix T into an upper (lower) Hessenberg matrix H, requiring $O(n^2)$ finite precision operations.

Since the construction of the eigenvector with the proposed method relies only on the coefficients of the involved Givens rotations, we show that it is not needed to update the whole Hessenberg matrix H to compute the Givens coefficients, but only the elements close to the tridiagonal matrix, by describing the first two steps of one sweep of IQR with shift λ applied to T. These steps are graphically depicted in Figure 4, in which the gray area denotes the part of the matrix modified by the multiplication of a Givens rotation.

Let $T_0 = T$. Let $\mathbf{v} \in \mathbb{R}^n$ be the first column of $(T - \lambda I_n)(T - \bar{\lambda} I_n)$. Hence, only the first three entries of $\mathbf{v}$ are different from zero.

In the first step of IQR two Givens rotations $\check{G}_1^{(r)}$ and $\breve{G}_1^{(r)}$ are constructed such that

$$\check{G}_1^{(r)} \breve{G}_1^{(r)} \mathbf{v} = \mathbf{e}_1,$$

where $\mathbf{e}_1 \in \mathbb{R}^n$ is the first vector of the canonical basis of $\mathbb{R}^n$. Then, the first step ends applying the following similarity transformation to T :

$$\check{G}_1 \breve{G}_1^{(r)} T \breve{G}_1^{(r)^T} \check{G}_1^{(r)^T}. \tag{5}$$

The multiplication $T \leftarrow \breve{G}_1^{(r)} T$ (Fig. 4 (a) $\rightarrow$(b)) introduces a bulge in position $(3,1)$, while $T \leftarrow T \breve{G}_1^{(r)^T}$ (Fig. 4 (b) $\rightarrow$(c)) introduces a bulge in position $(4,2)$. Furthermore, $T \leftarrow \check{G}_1^{(r)} T$ and $T \leftarrow T \check{G}_1^{(r)^T}$ (Fig. 4 (c) $\rightarrow$(d) and Fig. 4 (d) $\rightarrow$(e), respectively) introduce a bulge in position $(4,1)$. We observe that in the latter operation the entries of the first row of T (depicted in blue in Fig. 4 (d)) do not play any role in the computation of the next Givens coefficients and can be discarded.

At the second step the Givens rotations $\check{G}_2^{(r)}$ and $\breve{G}_2^{(r)}$ are applied to T in order to restore the Hessenberg structure, annihilating the entries $(4,1)$ and $(3,1)$ (denoted by "$\otimes$" in Fig. 4 (e) and (g)). Since $\check{G}_2^{(r)}$ and $\breve{G}_2^{(r)}$ act on the

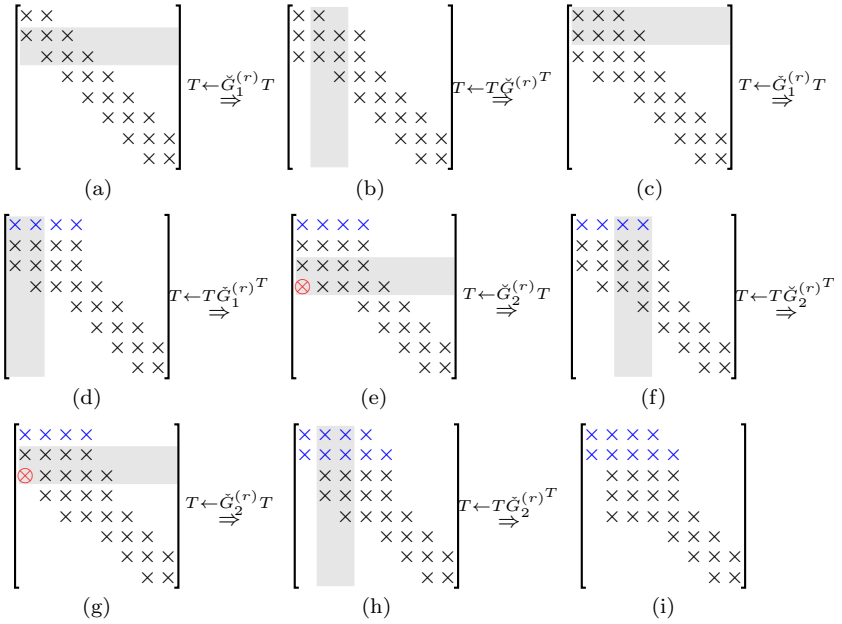

Fig. 4 Graphical description of the first two steps of a sweep of IQR with shift λ.

4th and the 3rd row and on the 3rd and 2nd row, respectively, the bulges in positions $(4,2),(5,2)$ and $(5,3)$ are introduced (Fig. 4 (i)). We observe that, after the multiplication $T \leftarrow \check{G}_2^{(r)} T$, the second row of T (depicted in blue in Fig. 4 (d)) do not play any role in the computation of the other Givens coefficients and can be discarded.

Summarizing, in exact arithmetic, at each step of IQR only few entries of the matrix T need to be updated to compute the sequences of Givens rotations $\{\check{G}_i^{(r)}\}_{i=1}^{n-1}$ and $\{\check{G}_i^{(r)}\}_{i=1}^{n-1}$ with a number of operations depending linearly on n, the order of the matrix T. In a similar fashion, at each step of the IQL method with double shift λ and $\bar{\lambda}$, only few entries of the matrix T need to be updated in order to compute the two sequences of Givens rotations $\{\check{G}_i^{(l)}\}_{i=1}^{n-1}$ and $\{\check{G}_i^{(l)}\}_{i=1}^{n-1}$ that transform the matrix into a similar lower block Hessenberg one,

$$\begin{bmatrix} H_1^{(l)} & \\ F & H_2^{(l)} \end{bmatrix},$$

with $H_1^{(l)} \in \mathbb{R}^{2\times 2}$ and $H_2^{(l)} \in \mathbb{R}^{(n-2)\times(n-2)}$ both Hessenberg and λ and $\bar{\lambda}$ are the eigenvalues of $H_1^{(l)}$. The left eigenvector $\mathbf{y}$ associated to λ can be retrieved either from the last two columns of $Q^{(r)}$ or from the first to columns of $Q^{(l)}$:

$$\mathbf{y} \equiv \mathbf{y}^{(r)} \equiv \mathbf{y}^{(l)},$$

with

$$\mathbf{y}^{(r)} = Q^{(r)}(:, n-1) + i \cdot Q^{(r)}(:, n), \quad \mathbf{y}^{(l)} = Q^{(l)}(:, 1) + i \cdot Q^{(l)}(:, 2), \tag{6}$$

and

$$Q^{(r)} = \check{G}_1^{(r)^T} \check{G}_1^{(r)^T} \cdots \check{G}_{n-1}^{(r)^T} \check{G}_{n-1}^{(r)^T},$$
$$Q^{(l)} = \check{G}_1^{(l)^T} \check{G}_1^{(l)^T} \cdots \check{G}_{n-1}^{(l)^T} \check{G}_{n-1}^{(l)^T}.$$

In finite precision arithmetic, similarly to the case of complex shift analyzed in Section 3, forward instability can occur in the computation of the latter sequences. The index j can be retrieved from the sequences $\check{c}_i^{(r)},\ i = 1, \ldots, n-1$, and $\check{c}_i^{(l)},\ i = 1, \ldots, n-1$, generated by one sweep of the IQR and IQL with double shift λ and $\bar{\lambda}$. We omit the details for the sake of brevity.

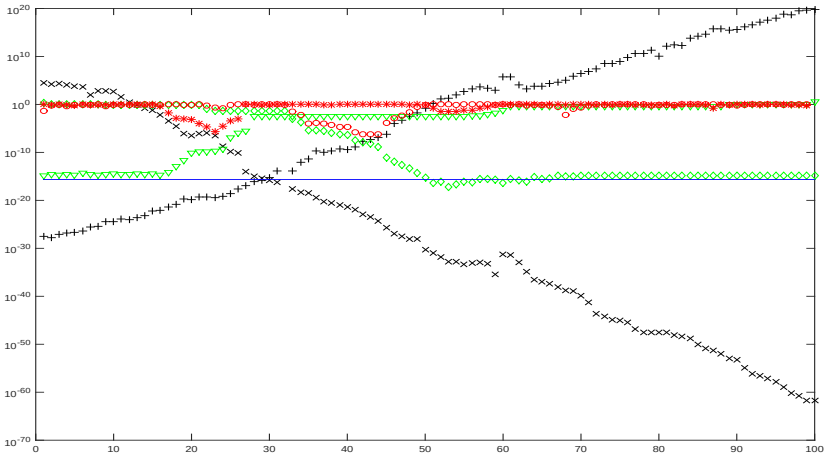

Fig. 5 Example 1 : plot of $|\mathbf{y} - \tilde{\mathbf{y}}^{(r)}|$ ("×"), $|\mathbf{y} - \tilde{\mathbf{y}}^{(l)}|$ ("+"), $\{\hat{\sigma}_i\}_{i=1}^n$ ("◇"), $\{\tilde{\sigma}_i\}_{i=1}^n$ ("▽"), $\{\check{c}_i^{(r)}\}_{i=1}^{n-1}$ ("○") and $\{\check{c}_i^{(l)}\}_{i=1}^{n-1}$("∗"), the machine precision ε_M (blue line) in logarithmic scale.

We apply IQR and IQL with double shift to the data of Example 1. In particular, denoted respectively by $\tilde{\mathbf{y}}^{(r)}$ and $\tilde{\mathbf{y}}^{(l)}$ the eigenvectors (6) computed in finite precision arithmetic, in Figure 5 the component–wise errors of $|\mathbf{y} - \tilde{\mathbf{y}}^{(r)}|$ (denoted by "×") and of $|\mathbf{y} - \tilde{\mathbf{y}}^{(l)}|$ (denoted by "+"), the sequences $\{\hat{\sigma}_i\}_{i=1}^n$ (denoted by "◇") and $\{\tilde{\sigma}_i\}_{i=1}^n$ (denoted by "○") and $\{\check{c}_i^{(r)}\}_{i=1}^{n-1}$ (denoted "○"), $\{\check{c}_i^{(l)}\}_{i=1}^{n-1}$("∗"), the machine precision ε_M (denoted by a blue line) are depicted on a logarithmic scale. The behaviour of the vectors and sequences depicted in Figure 5 is the same of the corresponding ones described in Example 1.

6 Numerical examples

All the numerical experiments of this section are performed in `MATLAB Ver. 2014b`, with machine precision $\varepsilon_M \sim 2.22 \times 10^{-16}$.

Example 2 In this example we consider a nonsymmetric tridiagonal matrix $T_n \in \mathbb{R}^{n\times n}$, $n = 200$, whose elements are generated by the `MATLAB` function `randn`.

We computed the eigenvectors $\hat{\mathbf{y}}_i$, $i = 1, \ldots, n$, by the described method and then recomputed the corresponding eigenvalues as the Rayleigh quotients

$$\hat{\lambda}_i = \frac{\hat{\mathbf{y}}_i^H T \hat{\mathbf{y}}_i}{\hat{\mathbf{y}}_i^H \hat{\mathbf{y}}_i}, \quad , i = 1, \ldots, n.$$

In Figure 6 the moduli of the differences $|\hat{\lambda}_i - \lambda_i|$, $i = 1, \ldots, n$, where λ_i are the eigenvalues computed by `eig` of `matlab`, are depicted. We can observe that such differences are of the order of ε_M.

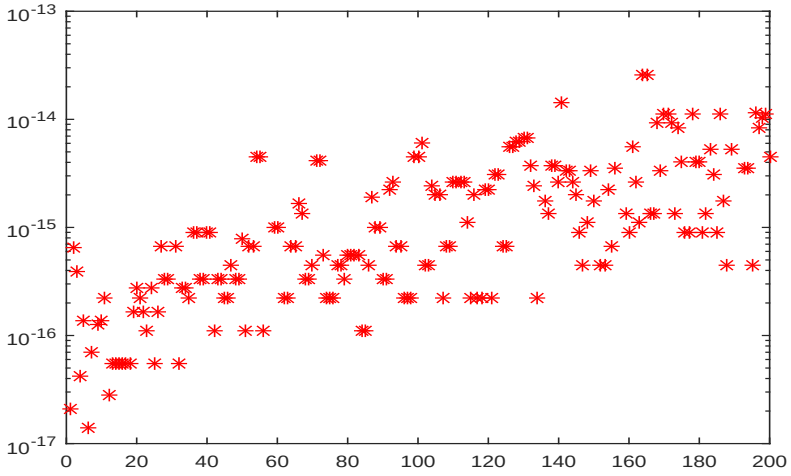

Fig. 6 Error, in logarithmic scale, of $|\hat{\lambda}_i - \lambda_i|$, $i = 1, \ldots, n$ (Example 2).

For each eigenvector $\tilde{\mathbf{y}}_i^{(1)}$, $i = 1, \ldots, n$, computed with the proposed method, denoted by

$$\nu_i = \|\tilde{\mathbf{y}}_i^{(1)^H} T - (\tilde{\mathbf{y}}_i^{(1)^H} T \tilde{\mathbf{y}}_i^{(1)}) \tilde{\mathbf{y}}_i^{(1)^H}\|_2$$

the norm of the ith residual, it turns out that

$$1.16 \times 10^{-13} = \max_k \nu_k \leq \nu_i \leq \min_k \nu_k = 2.18 \times 10^{-16}.$$

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

Example 3 In this example we consider the Bessel matrix

$$B_n^{(a,b)} = \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \gamma_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix},$$

with $n = 50$ and

$$\begin{array}{ll} \alpha_1 = -\frac{b}{a} & \alpha_j = -b\frac{a-2}{(2j+a-2)(2j+a-4)},\ j = 2,\dots,n, \\ \beta_1 = -\frac{\alpha_1}{a+1} & \alpha_j = -b\frac{j}{(2j+a-1)(2j+a-2)},\ j = 2,\dots,n-1, \\ \gamma_1 = -\alpha_1 & \gamma_j = b\frac{j+a-2}{(2j+a-2)(2j+a-3)}, \quad j = 2,\dots,n-1. \end{array}$$

The eigenvalues of these matrices $B_n^{(a,b)}$ suffer from ill–conditioning increasing with n [6]. As in [6], we take $a = -4.5$ and $b = 2$ and denote $B_{50}^{(-4.5,2)}$ simply by B. This matrix has well separated complex eigenvalues.

The matrix B and its eigenvalues $\lambda_i,\ i = 1,\dots,50$, are first computed in `matlab` using the variable–precision arithmetic with 128 significant digits, and then rounded to double precision, obtaining $\tilde{B}$ and $\tilde{\lambda}_i,\ i = 1,\dots,50$.

Let $\hat{\lambda}_i,\ i = 1,\dots,50$, be the eigenvalues of $\tilde{B}$ yielded by `eig` of `matlab`. In Figure 7, $\{\tilde{\lambda}_i\}_{i=1}^{50}$ and $\{\hat{\lambda}_i\}_{i=1}^{50}$, denoted respectively by "○" and "×", are depicted. Observe that $\tilde{\lambda}_i$ and $\hat{\lambda}_i$ are very different, since, as mentioned above, the eigenvalues of B suffer from ill-conditioning that increases with n [6].

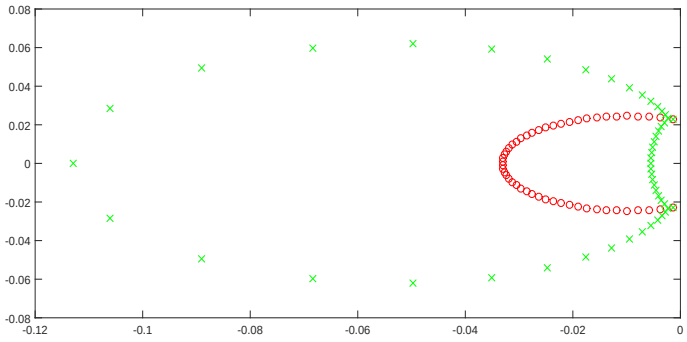

Fig. 7 Plot of the eigenvalues of B of Example 2 denoted by "○" and those computed by `eig` of `matlab`, denoted by "×".

The left eigenvectors associated to the eigenvalues $\tilde{\lambda}_i$ of $\tilde{B}$ are then computed with the proposed method, and denoted by $\tilde{\mathbf{y}}_i^{(1)}$, and as the last column

of the Q factor of the QR factorization of $\tilde{B}-\tilde{\lambda}_i I$, and denoted by $\tilde{\mathbf{y}}_i^{(2)}$. Hence, new approximations of the eigenvalues of $\tilde{B}$ are computed as the Rayleigh quotients :

$$\tilde{\lambda}_i^{(1)} = \tilde{\mathbf{y}}_i^{(1)^T} \tilde{B} \tilde{\mathbf{y}}_i^{(1)}, \quad \tilde{\lambda}_i^{(2)} = \tilde{\mathbf{y}}_i^{(2)^T} \tilde{B} \tilde{\mathbf{y}}_i^{(2)}, \quad i = 1, \ldots, 50.$$

The eigenvalues $\tilde{\lambda}_i$ and the computed approximations, $\tilde{\lambda}_i^{(1)}$ and $\tilde{\lambda}_i^{(2)}, i = 1, \ldots, 50$, of the eigenvalues of $\tilde{B}$, are depicted in Figure 8 and denoted respectively by the symbols "$\circ$", "∇" and "$+$".

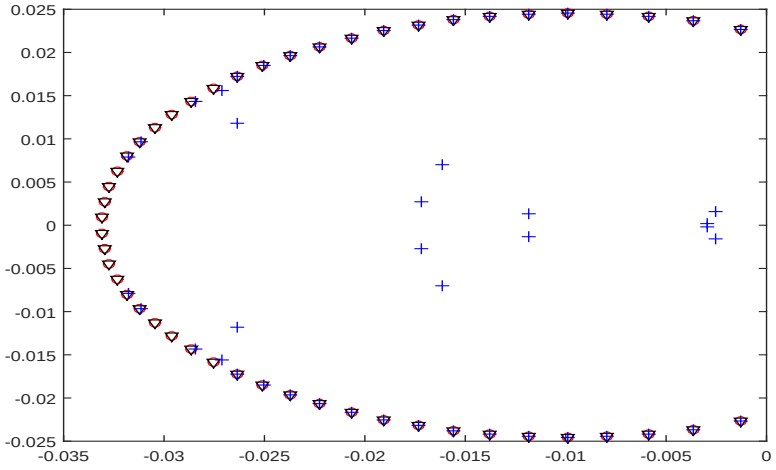

Fig. 8 Example 2. Plot of $\tilde{\lambda}_i$ $\tilde{\lambda}_i^{(1)}$ and $\tilde{\lambda}_i^{(2)}, i = 1, \ldots, 50$, denoted respectively by "$\circ$", "∇" and "$+$".

Observe that $\tilde{\lambda}_i$ and $\tilde{\lambda}_i^{(1)}$ perfectly overlap, while $\tilde{\lambda}_i^{(2)}$ departs from $\tilde{\lambda}_i$ as the negative imaginary part increases.

In Table 2 are reported $\max_i |\tilde{\lambda}_i - \tilde{\lambda}_i^{(1)}|$ and $\max_i |\tilde{\lambda}_i - \tilde{\lambda}_i^{(2)}|$. We observe

$\max_i \lvert\tilde{\lambda}_i - \tilde{\lambda}_i^{(1)}\rvert$	$\max_i \lvert\tilde{\lambda}_i - \tilde{\lambda}_i^{(2)}\rvert$
3.06×10^{-15}	3.01×10^{2}

Table 2 $\max_i |\tilde{\lambda}_i - \tilde{\lambda}_i^{(1)}|$ and $\max_i |\tilde{\lambda}_i - \tilde{\lambda}_i^{(2)}|$.

that, if we compute an approximation of the eigenvector $\breve{\mathbf{y}}$ of one of the eigenvalues $\tilde{\lambda}_i$ of $\tilde{B}$ by inverse iteration, the vector $\breve{\mathbf{y}}$ is very far from $\tilde{\mathbf{y}}_i^{(1)}$. This behavior is described in [12]. Furthermore, the corresponding Raylegh quotient $\breve{\mathbf{y}}^H \tilde{B} \breve{\mathbf{y}} / (\breve{\mathbf{y}}^H \breve{\mathbf{y}})$ is far from the eigenvalue $\tilde{\lambda}_i$.

For each eigenvector $\tilde{\mathbf{y}}_i^{(1)}$, $i = 1, \ldots, 50$, computed with the proposed method, denoted by

$$\nu_i = \|\tilde{\mathbf{y}}_i^{(1)^H} \tilde{B} - (\tilde{\mathbf{y}}_i^{(1)^H} \tilde{B} \tilde{\mathbf{y}}_i^{(1)}) \tilde{\mathbf{y}}_i^{(1)^H}\|_2,$$

the norm of the ith residual, it turns out that

$$3.06 \times 10^{-15} = \max_k \nu_k \leq \nu_i \leq \min_k \nu_k = 2.61 \times 10^{-16}.$$

Example 4 In this example we consider the Clement matrix of order $n = 200$ [1], also called Sylvester–Kac matrix [3],

$$C_n = \begin{bmatrix} & \gamma_1 & & & \\ \beta_1 & & \gamma_2 & & \\ & \beta_2 & & \ddots & \\ & & \ddots & & \gamma_{n-1} \\ & & & \beta_{n-1} & \end{bmatrix},$$

with $\gamma_i = i$ and $\beta_i = n - i$, $i = 1, \ldots, n-1$, whose eigenvalues are [3] $\lambda_k = n - 2(k-1)$, for $k = 1, \ldots, n$, and the corresponding left eigenvector matrix [3] is

$$u_{k,j} = \sum_{i=0}^{\min(k-1, j-1)} (-1)^i \binom{j-1}{i} \binom{n-j-1}{k-i-1}.$$

Also in this example, for each eigenvector $\tilde{\mathbf{y}}_i^{(1)}$, $i = 1, \ldots, n$ computed with the proposed method, denoted by

$$\nu_i = \|\tilde{\mathbf{y}}_i^{(1)^H} C_n - (\tilde{\mathbf{y}}_i^{(1)^H} C_n \tilde{\mathbf{y}}_i^{(1)}) \tilde{\mathbf{y}}_i^{(1)^H}\|_2,$$

the norm of the ith residual, it turns out that

$$5.67 \times 10^{-13} = \max_k \nu_k \leq \nu_i \leq \min_k \nu_k = 4.49 \times 10^{-16}.$$

7 Conclusions

The problem of computing the eigenvector associated to a computed eigenvalue λ of a nonsymmetric tridiagonal matrix T is considered in this manuscript. It is first described how the left eigenvector $\mathbf{y}$ associated to an eigenvalue λ of T can be obtained applying one sweep of the QR or the QL method with shift λ. Unfortunately, both methods can suffer from forward instability when working in finite precision arithmetic.

We propose a method based on computing an index j that combines the eigenvector yielded by one sweep of the QR and one sweep of the QL methods with shift λ avoiding the instability phenomena. The index j is computed considering the two sequences of cosines generated by a sweep of QR and

QL method with shift λ. The sought eigenvector is obtained from the first j Givens coefficients generated by the QR method and from the first $n-j$ Givens coefficients generated by the QL method.

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

The results of the numerical examples look very promising showing that the eigenvectors are computed with high accuracy.

References

1. Bini, D., Gemignani, L., Tisseur, F.: The Ehrlich-Aberth method for the nonsymmetric tridiagonal eigenvalue problem. SIAM Journal on Matrix Analysis and Applications **27**(1), 153–175 (2005)
2. Chang, X.W., Paige, C.C., Stewart, G.W.: Perturbation analyses for the QR factorization. SIAM J. Matrix Anal. Appl. **18**(3), 775–791 (1997)
3. Chu, W., Wang, X.: Eigenvectors of tridiagonal matrices of Sylvester type. Calcolo **45**(4), 217–233 (2008)
4. Dax, A., Kaniel, S.: The ELR method for computing the eigenvalues of a general matrix. SIAM Journal on Numerical Analysis **18**(4), 597–605 (1981)
5. Dongarra, J.J., Geist, G.A., Romine, C.H.: Algorithm 710: Fortran subroutines for computing the eigenvalues and eigenvectors of a general matrix by reduction to general tridiagonal form. ACM Trans. Math. Softw. **18**(4), 392–400 (1992)
6. Ferreira, C., B. Parlett, B., Dopico, F.: Sensitivity of eigenvalues of an unsymmetric tridiagonal matrix. Numerische Mathematik **122**(3), 527–555 (2012)
7. Francis, J.: The QR transformation. a unitary analogue to the LR transformation. I. Comput. J. **4**, 265–271 (1961)
8. Francis, J.: The QR transformation. a unitary analogue to the LR transformation. II. Comput. J. **4**, 332–345 (1961)
9. Freund, R.W., Gutknecht, M.H., Nachtigal, N.M.: An implementation of the lookahead Lanczos algorithm for non-Hermitian matrices. SIAM J. Sci. Comput. **14**, 137–158 (1993)
10. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th ed. Johns Hopkins University Press, Baltimore (2013)
11. Horn, R., Johnson, C.: Topics in Matrix Analysis. Cambridge University Press, New York (1991)
12. Ipsen, I.C.F.: Computing an eigenvector with inverse iteration. SIAM Review **39**(2), 254–291 (1997)
13. Mastronardi, N., Taeter, H., Dooren, P.V.: On Computing Eigenvectors of Symmetric Tridiagonal Matrices, vol. 30, pp. 181–195. Bini D., Di Benedetto F., Tyrtyshnikov E., Van Barel M. (eds), Springer INdAM Series, Cham (2019)
14. Parlett, B.N., Le, J.: Forward instability of tridiagonal QR. SIAM J. Matrix Anal. Appl. **14**(1), 279–316 (1993)
15. Pasquini, L.: Accurate computation of the zeros of the generalized Bessel polynomials. Numerische Mathematik **86**(3), 507–538 (2000)
16. Rutishauser, H.: Solution of Eigenvalue Problems with the LR-transformation. Nat. Bur. Standards **49**, 47–81 (1958)
17. Sidje, R.B., Burrage, K.: QRT: A QR-based tridiagonalization algorithm for nonsymmetric matrices. SIAM Journal on Matrix Analysis and Applications **26**(3), 878–900 (2005)
18. Wang, H.H., Gregory, R.T.: On the reduction of an arbitrary real square matrix to tridiagonal form. Math. Comp. **18**, 501–505 (1964)
19. Watkins, D.S.: The transmission of shifts and shift blurring in the QR algorithm. Linear Algebra and its Applications **241–243**, 877–896 (1996)