



Computing Gaussian quadrature rules with high relative accuracy

Teresa Laudadio¹ · Nicola Mastronardi¹ · Paul Van Dooren²

Received: 24 November 2021 / Accepted: 7 March 2022 / Published online: 2 June 2022
© The Author(s) 2022

Abstract

The computation of n -point Gaussian quadrature rules for symmetric weight functions is considered in this paper. It is shown that the nodes and the weights of the Gaussian quadrature rule can be retrieved from the singular value decomposition of a bidiagonal matrix of size $n/2$. The proposed numerical method allows to compute the nodes with high relative accuracy and a computational complexity of $\mathcal{O}(n^2)$. We also describe an algorithm for computing the weights of a generic Gaussian quadrature rule with high relative accuracy. Numerical examples show the effectiveness of the proposed approach.

Keywords Gaussian quadrature rule · Golub and Welsch algorithm · Singular value decomposition

1 Introduction

The Golub and Welsch algorithm [12] is the classical way to compute the nodes λ_i and the weights ω_i of an n -point Gaussian quadrature rule (GQR), commonly used

Teresa Laudadio, Nicola Mastronardi and Paul Van Dooren contributed equally to this work.

Dedicated to Claude Brezinski on the occasion of his 80th birthday.

✉ Teresa Laudadio
teresa.laudadio@cnr.it

Nicola Mastronardi
nicola.mastronardi@cnr.it

Paul Van Dooren
paul.vandooren@uclouvain.be

¹ Istituto per le Applicazioni del Calcolo “M. Picone”, Consiglio Nazionale delle Ricerche, via Amendola 122/D, Bari, Italy

² Department of Mathematical Engineering, Catholic University of Louvain, Avenue Georges Lemaitre 4, Louvain-la-Neuve, Belgium

to approximate integrals of the following kind,

$$\int_{-\tau}^{\tau} f(x)\omega(x)dx \approx \sum_{i=1}^n f(\lambda_i)\omega_i$$

where $\omega(x) \geq 0$ is a positive weight function in the interval $[-\tau, \tau]$, with $\tau > 0$ and $f(x)$ is a continuous function in the same interval. It computes the nodes, also called *knots*, as the eigenvalues of a symmetric tridiagonal matrix of order n , called the Jacobi matrix, while the corresponding weights are easily obtained from the first components of the associated eigenvectors (see [12] for more details). The nonzero entries of the Jacobi matrix are computed from the coefficients of the three-term recurrence relation of the orthogonal polynomials associated with the weight function $\omega(x)$.

The Matlab function `gauss` is a straightforward implementation of the Golub and Welsch algorithm [12], included in the package “OPQ: A Matlab suite of programs for generating orthogonal polynomials and related quadrature rules” by Walter Gautschi, and it is available at <https://www.cs.purdue.edu/archives/2002/wxg/codes/OPQ.html>.

In [19] it is shown that, when the weight function $\omega(x)$ is symmetric with respect to the origin, the positive nodes of the n -point GQR are the eigenvalues of a symmetric tridiagonal matrix, of order $\frac{n}{2}$ if n is even and $\frac{n+1}{2}$ if n is odd, obtained by eliminating half of the unknowns from the original eigenvalue problem. Moreover, the Cholesky factor of the latter tridiagonal matrix of reduced order is a nonnegative lower bidiagonal matrix of dimensions $\frac{n}{2} \times \frac{n}{2}$, if n is even, and $\frac{n+1}{2} \times \frac{n-1}{2}$, if n is odd, whose entries in its main diagonal and lower subdiagonal are respectively the odd and the even elements of the first subdiagonal of the original Jacobi matrix.

In the same paper, the authors propose modifications of the Golub and Welsch algorithm for computing the nodes and weights of a symmetric GQR (SGQR), by means of a three-term recurrence relation corresponding to the orthogonal polynomials associated with the modified weight function. This resulted in faster Matlab functions than `gauss`.

Inspired by [19], in this paper we propose to compute the positive nodes of the n -point SGQR as the singular values of a nonnegative bidiagonal matrix. These can be computed in $\mathcal{O}(n^2)$ floating point operations and with high relative accuracy by the algorithm described in [5, 6]. Moreover, the weights of the SGQR can be obtained from the first row of the right singular vector matrix. The stability of the three-term recurrence relation arising in computing the weights of a nonsymmetric n -point GQR is also analyzed and a novel numerical method for computing the weights with high relative accuracy is proposed.

A different approach for computing nodes and weights of a GQR in $\mathcal{O}(n)$ operations has been introduced in [10] and is implemented in the Matlab package `Chebfun` [15]. This method is based on approximations of the nodes and weights that appear to be relatively accurate in most nodes but no proof of such a result is given. For a very large number of nodes, this method is today the most efficient one [14, 25].

The paper is organized as follows. The basic notation and definitions used in the paper are listed in Section 2. In Section 3, the main features of the Golub and Welsch algorithm are described. In Section 4, the proposed algorithm for computing the nodes of an n -point SGQR as the singular values of a bidiagonal matrix is described. Different techniques for computing the weights of an n -point GQR are described in Section 5. In Section 6 we show the elementwise relative accuracy of the weights computed by our new method and in Section 7, we give a number of numerical examples confirming our stability results. We then end with a section of concluding remarks.

2 Notation and definitions

Matrices are denoted by upper-case letters $X, Y, \dots, \Lambda, \Theta, \dots$; vectors with bold lower-case letters $\mathbf{x}, \mathbf{y}, \dots, \boldsymbol{\lambda}, \boldsymbol{\theta}, \dots$; scalars with lower-case letters $x, y, \dots, \lambda, \theta, \dots$. The element i, j of a matrix A is denoted by a_{ij} and the i th element of a vector \mathbf{x} is denoted by x_i .

Submatrices are denoted by the colon notation of Matlab: $A(i : j, k : l)$ denotes the submatrix of A formed by the intersection of rows i to j and columns k to l , and $A(i : j, :)$ and $A(:, k : l)$ denote the rows of A from i to j and the columns of A and from k to l , respectively. Sometimes, $A(i : j, k : l)$ is also denoted by $A_{i:j,k:l}$ and $\mathbf{x}(i : j)$ by $x_{i:j}$.

The identity matrix of order n is denoted by I_n , and its i th column, $i = 1, \dots, n$, i.e., the i th vector of the canonical basis of \mathbb{R}^n , is denoted by \mathbf{e}_i . The notation $\lfloor y \rfloor$ stands for the largest integer not exceeding $y \in \mathbb{R}_+$.

Given a vector $\mathbf{x} \in \mathbb{R}^n$, then $\text{diag}(\mathbf{x})$ is a diagonal matrix of order n , with the elements of vector \mathbf{x} on the main diagonal. Given a matrix $A \in \mathbb{R}^{m \times n}$, then $\text{diag}(A)$ is a column vector of length $i = \min(m, n)$, whose entries are those of the main diagonal of A .

3 Computing the zeros of orthogonal polynomials

Let $p_\ell(x) = k_\ell x^\ell + \sum_{j=0}^{\ell-1} c_j x^j$, $\ell = 0, 1, \dots$, be the sequence of orthonormal polynomials with respect to a positive weight function $\omega(x)$ in the interval $[-\tau, \tau]$, $\tau > 0$, i.e.,

$$\int_{-\tau}^{\tau} p_i(x)p_j(x)\omega(x)dx = \delta_{ij}, \quad \text{with } \delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases}$$

where k_ℓ is chosen to be positive. The polynomials $p_\ell(x)$ satisfy the following three-term recurrence relation [23]

$$\begin{cases} p_{-1}(x) = 0, \\ p_0(x) = k_0 = \frac{1}{\sqrt{\mu_0}}, \\ \gamma_{\ell+1}p_{\ell+1}(x) = (x - \theta_\ell)p_\ell(x) - \gamma_\ell p_{\ell-1}(x), \quad \ell \geq 0, \end{cases} \tag{1}$$

where

$$\mu_0 = \int_{-\tau}^{\tau} \omega(x) dx,$$

and

$$\begin{cases} \gamma_0 = 0 \\ \gamma_\ell = k_\ell/k_{\ell-1} > 0 \end{cases}, \quad \theta_\ell = \frac{\int_{-\tau}^{\tau} x p_\ell^2(x) \omega(x) dx}{\int_{-\tau}^{\tau} p_\ell^2(x) \omega(x) dx}, \quad \ell = 1, 2, \dots$$

Using (1), we can write the n -step recurrence relation

$$J\mathbf{p}(x) = x\mathbf{p}(x) - \gamma_n p_n(x)\mathbf{e}_n,$$

where

$$J = \begin{bmatrix} \theta_0 & \gamma_1 & & & \\ \gamma_1 & \theta_1 & \gamma_2 & & \\ & \gamma_2 & \ddots & \ddots & \\ & & \ddots & \theta_{n-2} & \gamma_{n-1} \\ & & & \gamma_{n-1} & \theta_{n-1} \end{bmatrix}, \quad \mathbf{p}(x) = \begin{bmatrix} p_0(x) \\ p_1(x) \\ \vdots \\ p_{n-2}(x) \\ p_{n-1}(x) \end{bmatrix}. \quad (2)$$

The matrix J is called the Jacobi matrix [7]. The following theorem was shown in [7, Th. 1.31].

Theorem 1 Let $J = Q\Lambda Q^T$ be the spectral decomposition of J , where $\Lambda \in \mathbb{R}^{n \times n}$ is a diagonal matrix and

$$\boldsymbol{\lambda} := \text{diag}(\Lambda) = [\lambda_1, \dots, \lambda_n]^T, \quad \lambda_1 < \lambda_2 < \dots < \lambda_n,$$

and $Q \in \mathbb{R}^{n \times n}$ is an orthogonal matrix, i.e., $Q^T Q = I_n$. Then $p_n(\lambda_i) = 0$, $i = 1, \dots, n$, and $Q = VD$, with

$$V = [\mathbf{p}(\lambda_1), \mathbf{p}(\lambda_2), \dots, \mathbf{p}(\lambda_n)], \quad D = \begin{bmatrix} \hat{\omega}_1 & & & \\ & \hat{\omega}_2 & & \\ & & \ddots & \\ & & & \hat{\omega}_n \end{bmatrix}, \quad (3)$$

and

$$\hat{\omega}_i = \frac{1}{\|\mathbf{p}(\lambda_i)\|_2} = \frac{1}{\sqrt{\sum_{\ell=0}^{n-1} p_\ell^2(\lambda_i)}}.$$

The eigenvalues λ_i are the nodes of the GQR and the corresponding weights ω_i are defined as (see [23])

$$\omega_i := \omega_i(\lambda_i) = \hat{\omega}_i^2 = \frac{1}{\sum_{\ell=0}^{n-1} p_\ell^2(\lambda_i)}, \quad i = 1, \dots, n. \quad (4)$$

Hence, the weights ω_i of the GQR can be determined by (4), computing $p_\ell(\lambda_i)$, $\ell = 0, 1, \dots, n-1$, $i = 1, \dots, n$, either by means of the three-term recurrence relation (1), as done in [19], or by computing the whole eigenvector matrix Q of J in (3).

Both approaches will be described in Section 5, and their stability analysis will be provided as well.

As shown in [26], the weights can be also obtained by the first row of Q as

$$\omega_i = \mu_{0q1,i}^2, \quad i = 1, \dots, n. \tag{5}$$

The Golub and Welsch algorithm [12], relying on a modification of the QR -method [4], yields the nodes and the weights of the GQR by computing only the eigenvalues of the Jacobi matrix and the first row of the associated eigenvector matrix.

3.1 Computing the zeros of orthogonal polynomials for a symmetric weight function

For a weight function $\omega(x)$, symmetric with respect to the origin, the diagonal elements θ_ℓ , $\ell = 1, \dots, n$, in (2) become zero since, as shown in [7, Th. 1.17],

$$p_\ell(-x) = (-1)^\ell p_\ell(x), \quad \ell = 0, 1, 2, \dots, \tag{6}$$

and, thus, $x p_\ell^2(x)$ is an odd function in $[-\tau, \tau]$. Therefore, the Jacobi matrix in (2) becomes

$$J = \begin{bmatrix} 0 & \gamma_1 & & & \\ \gamma_1 & 0 & \gamma_2 & & \\ & \gamma_2 & \ddots & \ddots & \\ & & \ddots & 0 & \gamma_{n-1} \\ & & & \gamma_{n-1} & 0 \end{bmatrix}. \tag{7}$$

Furthermore, by (6), if λ_i , $i = 1, \dots, n$, is a zero of p_ℓ , $\ell \geq 0$, i.e., $p_\ell(\lambda_i) = 0$, then $\lambda_{n-i+1} = -\lambda_i$ is also a zero of p_ℓ , since

$$p_\ell(\lambda_{n-i+1}) = p_\ell(-\lambda_i) = (-1)^\ell p_\ell(\lambda_i) = 0.$$

As a consequence,

$$\omega_i = \frac{1}{\sum_{\ell=0}^{n-1} p_\ell^2(\lambda_i)} = \frac{1}{\sum_{\ell=0}^{n-1} p_\ell^2(\lambda_{n-i+1})} = \omega_{n-i+1}, \quad i = 1, \dots, n.$$

Therefore, for weight functions symmetric with respect to the origin, it is sufficient to compute only the positive nodes and the corresponding weights.

Without loss of generality, in the sequel we will consider the following reordering of λ , Λ and V , i.e.,

$$\lambda := \lambda(\mathbf{j}), \quad \Lambda := \Lambda(\mathbf{j}, \mathbf{j}), \quad V := V(:, \mathbf{j}),$$

where \mathbf{j} is the following permutation of the index set $i = [1, \dots, n]$

$$\mathbf{j} = \begin{cases} [n, n-1, \dots, \frac{n}{2} + 1, 1, 2, \dots, \frac{n}{2}], & \text{if } n \text{ even,} \\ [n, n-1, \dots, \frac{n+3}{2}, 1, 2, \dots, \frac{n-1}{2}, \frac{n+1}{2}], & \text{if } n \text{ odd.} \end{cases}$$

Hence, the first $\lfloor \frac{n}{2} \rfloor$ entries of λ are the strictly positive eigenvalues of J in a decreasing order, the second $\lfloor \frac{n}{2} \rfloor$ entries of λ are the negative eigenvalues of J in an increasing order, and, if n is odd, the last entry of λ is the zero eigenvalue.

4 Computation of the nodes and the weights of a symmetric Gaussian quadrature rule

In this section we propose an alternative method to compute the positive nodes and corresponding weights of a SGQR. We will need the following Lemma [11].

Lemma 1 *Let $A \in \mathbb{R}^{m \times n}$, $m \leq n$, and*

$$S_1 = A^T A, \quad S_2 = AA^T.$$

Let $A = U\Sigma V^T$ be the singular value decomposition of A , with $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ orthogonal, and

$$\Sigma = [\text{diag}(\sigma_1, \sigma_2, \dots, \sigma_m) \ 0_{m, n-m}] \in \mathbb{R}^{m \times n},$$

where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m \geq 0$. Then

$$A^T A = V \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_m^2, \underbrace{0, \dots, 0}_{n-m}) V^T,$$

$$AA^T = U \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_m^2) U^T.$$

Moreover, if

$$V = [\underbrace{V_1}_m \ \underbrace{V_2}_{n-m}],$$

then

$$\begin{bmatrix} A & \\ & A^T \end{bmatrix} = W \Lambda W^T,$$

with

$$W = \frac{1}{\sqrt{2}} \begin{bmatrix} U & U & 0 \\ V_1 & -V_1 & \sqrt{2}V_2 \end{bmatrix}$$

and

$$\Lambda = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_m, -\sigma_1, -\sigma_2, \dots, -\sigma_m, \underbrace{0, \dots, 0}_{n-m}).$$

The following Corollary holds.

Corollary 1 Let B_n be the bidiagonal matrix

$$B = \begin{cases} \begin{bmatrix} \gamma_1 & \gamma_2 & & & \\ & \gamma_3 & \gamma_4 & & \\ & & \gamma_5 & \ddots & \\ & & & \ddots & \gamma_{n-2} \\ & & & & \gamma_{n-1} \end{bmatrix} \in \mathbb{R}^{\frac{n}{2} \times \frac{n}{2}} & \text{if } n \text{ even,} \\ \begin{bmatrix} \gamma_1 & \gamma_2 & & & \\ & \gamma_3 & \gamma_4 & & \\ & & \gamma_5 & \ddots & \\ & & & \ddots & \gamma_{n-3} \\ & & & & \gamma_{n-2} & \gamma_{n-1} \end{bmatrix} \in \mathbb{R}^{\frac{n-1}{2} \times \frac{n+1}{2}} & \text{if } n \text{ odd,} \end{cases}$$

with singular value decomposition

$$B = U \Sigma V^T,$$

where U and V are orthogonal, $\sigma = [\sigma_1, \sigma_2, \dots, \sigma_{\lfloor \frac{n}{2} \rfloor}]^T$, and

$$\begin{cases} U \in \mathbb{R}^{\frac{n}{2} \times \frac{n}{2}}, & \Sigma = \text{diag}(\sigma) \in \mathbb{R}^{\frac{n}{2} \times \frac{n}{2}}, & V \in \mathbb{R}^{\frac{n}{2} \times \frac{n}{2}}, & \text{if } n \text{ even,} \\ U \in \mathbb{R}^{\frac{n-1}{2} \times \frac{n-1}{2}}, & \Sigma = [\text{diag}(\sigma), 0] \in \mathbb{R}^{\frac{n-1}{2} \times \frac{n+1}{2}}, & V \in \mathbb{R}^{\frac{n+1}{2} \times \frac{n+1}{2}}, & \text{if } n \text{ odd,} \end{cases}$$

and, if n is odd, $V = [V_1 | v_2]$, $V_1 \in \mathbb{R}^{\frac{n+1}{2} \times \frac{n-1}{2}}$, $v_2 \in \mathbb{R}^{\frac{n+1}{2}}$. Then

$$\begin{aligned} J &= Q \Lambda Q^T \\ &= \begin{cases} \frac{1}{2} P^T \begin{bmatrix} U & U \\ V & -V \end{bmatrix} \hat{\Lambda} \begin{bmatrix} U & U \\ V & -V \end{bmatrix}^T P, & \text{if } n \text{ even,} \\ \frac{1}{2} P^T \begin{bmatrix} U & U \\ V_1 & -V_1 & \sqrt{2} v_2 \end{bmatrix} \hat{\Lambda} \begin{bmatrix} U & U & \mathbf{0} \\ V_1 & -V_1 & \sqrt{2} v_2 \end{bmatrix}^T P, & \text{if } n \text{ odd,} \end{cases} \end{aligned} \tag{8}$$

and $\hat{\Lambda}$ is the square diagonal matrix

$$\hat{\Lambda} = \begin{cases} \text{diag}([\sigma; -\sigma]) & \text{if } n \text{ even,} \\ \text{diag}([\sigma; -\sigma; 0]) & \text{if } n \text{ odd,} \end{cases}$$

where $P := I_n(\tilde{j}, \cdot)$ is the even-odd permutation matrix corresponding to the even-odd permutation of the index set $\mathbf{i} = [1, \dots, n]$, i.e.,

$$\tilde{j} = \begin{cases} [2, 4, \dots, n-2, n, 1, 3, \dots, n-3, n-1]^T & \text{if } n \text{ even,} \\ [2, 4, \dots, n-3, n-1, 1, 3, \dots, n-2, n]^T & \text{if } n \text{ odd.} \end{cases}$$

Proof Obviously, we have

$$P J P^T = \left[\begin{array}{c|c} & B \\ \hline B^T & \end{array} \right].$$

and (8) follows then straightforwardly from Lemma 1. □

Taking the even-odd structure of P into account, we have $\mathbf{e}_1^T P^T = \mathbf{e}_{\lfloor \frac{n}{2} \rfloor + 1}^T$ and thus the first row of Q is given by

$$\mathbf{e}_1^T Q = \begin{cases} \frac{1}{\sqrt{2}} \mathbf{e}_1^T P^T \begin{bmatrix} U & U \\ V & -V \end{bmatrix} = \frac{1}{\sqrt{2}} [V(1, :) - V(1, :)], & \text{if } n \text{ even,} \\ \frac{1}{\sqrt{2}} \mathbf{e}_1^T P^T \begin{bmatrix} U & U & \mathbf{0} \\ V_1 & -V_1 & \sqrt{2} \mathbf{v}_2 \end{bmatrix} = \left[\frac{V_1(1, :)}{\sqrt{2}}, -\frac{V_1(1, :)}{\sqrt{2}}, \mathbf{v}_2(1) \right], & \text{if } n \text{ odd,} \end{cases}$$

and, hence, by (4) and (5), the weights ω_i are given by

$$\begin{cases} \omega_i = \frac{\mu_0}{2} [V(1, i)]^2, & i = 1, \dots, \lfloor \frac{n}{2} \rfloor, \\ \omega_{n-i+1} = \omega_i, \end{cases}$$

and, for n odd,

$$\omega_{\frac{n+1}{2}} = \frac{\mu_0}{2} \left[V \left(1, \frac{n+1}{2} \right) \right]^2.$$

Remark 1 Transforming the problem of computing the eigenvalues of J into the problem of computing the singular values of the nonnegative bidiagonal matrix B , allows us to compute the nodes of a SGQR with high relative accuracy by using the algorithm described in [5, 6] and implemented in LAPACK (subroutine `dlaasq1.f`) [1]. Note that `dlaasq1.f` computes the singular values of square bidiagonal matrices and for the n -point SGQR with odd n , the bidiagonal matrix B is rectangular of size $\lfloor \frac{n}{2} \rfloor \times \lfloor \frac{n+1}{2} \rfloor$. But one step of the QR iteration with zero shift, implemented as described in [5], is sufficient to transform B to $[\hat{B} | \mathbf{0}]$, with \hat{B} square bidiagonal of order $\lfloor \frac{n}{2} \rfloor$ to high relative accuracy.

On the other hand, the componentwise stability of computing ω_i , $i = 1, \dots, n$, from the first row of either Q or V , is not ensured if these matrices are computed by the QR algorithm. Indeed, these orthogonal matrices are computed in a normwise backward stable manner but not necessarily in a componentwise stable manner [27, p. 236], [16]. In Section 7, we show in the example of Fig. 2 that using the eigenvalue decomposition of J for computing the sequence (10), the computation of the small weights is not elementwise stable even though the eigenvectors are computed in a normwise backward stable manner [16]. For the sake of completeness, it is worth mentioning that a similar behavior is not observed for other classes of orthogonal polynomials, such as the Jacobi ones, since the weights do not have such a large range of values and, thus, the computation of GQR for such weights with the function `gauss` is as accurate as the method proposed in this paper.

5 Computation of the weights of the Gaussian quadrature rules

In this section we consider different techniques for computing the weights ω_i , $i = 1, \dots, n$, of an n -point GQR, relying only on the knowledge of the corresponding nodes λ_i for nonsymmetric weight functions. At the end of the section we will shortly describe how these techniques can be adapted to symmetric weight functions. As pointed out in Section 3, the nodes and the weights of the n -point Gaussian rule associated with a weight function $\omega(x)$, are the zeros λ_i , $i = 1, \dots, n$, of the orthogonal polynomial $p_n(x)$ of degree n and the corresponding weights are

$$\omega_i = \frac{1}{\sum_{\ell=0}^{n-1} p_\ell^2(\lambda_i)}, \quad i = 1, \dots, n. \tag{9}$$

The following methods have been proposed in the literature for computing the sequence

$$p_\ell(\lambda_i), \quad \ell = 0, 1, \dots, n - 1, \tag{10}$$

1. the eigenvalue decomposition [11];
2. the forward three-term recurrence relation (FTTR) [8];
3. the backward three-term recurrence relation (BTTR) [20, 24].

The computation of the weights by means of the FTTR was considered in [19] without providing any stability analysis.

Given $\tilde{p}_0(\bar{\lambda})$ and a zero $\bar{\lambda}$ of $p_n(\lambda)$, we denote by $\tilde{p}_1(\bar{\lambda}), \tilde{p}_2(\bar{\lambda}), \dots, \tilde{p}_{n-1}(\bar{\lambda})$, the sequence computed by means of (1) in a forward manner, i.e., by FTTR. Analogously, since $p_n(\bar{\lambda}) = 0$, we can set an arbitrary value for $\hat{p}_{n-1}(\bar{\lambda})$, and denote by $\hat{p}_{n-2}(\bar{\lambda}), \hat{p}_{n-3}(\bar{\lambda}), \dots, \hat{p}_1(\bar{\lambda}), \hat{p}_0(\bar{\lambda})$, the sequence computed using (1) in a backward fashion, i.e., by BTTR. The latter procedure is often referred to as the Miller’s backward recurrence algorithm [3, 8, 20] in the literature. It was originally proposed by J.C.P. Miller for the computation of tables of the modified Bessel function [3, page xvii].

The stability of the generic FTTR and BTTR, i.e., sequences generated by three-term recurrence relations not linked to orthogonal polynomials, was analyzed in [8, 20, 24], and in these papers the preference went to the BTTR for stability purposes.

For the weight functions depending on the parameters α and β listed in Table 1, we carried out extensive tests, choosing many different values of α and β , for computing the sequences $p_j(\lambda_i)$, $j = 0, 1, \dots, n - 1$, by means of FTTR and BTTR. FTTR works in an accurate way for some classes of weights and BTTR works in an accurate way for other classes of weights.

The results of these experiments are displayed in Table 1 in which we mention “Ok” if the method computes the sequence in an accurate way for all the considered values of α and β , and “No” if the method fails to compute the sequence in an accurate way for some values of α and β . In Fig. 1 (left) we show on a logarithmic scale, the results obtained for the Hermite polynomials of degree j , $j = 0, 1, \dots, 127$, in the largest zero of the Hermite polynomial of degree 128, with FTTR in extended precision (128 digits), with FTTR in double precision and with BTTR in double precision. While the Hermite polynomials are accurately computed with FTTR, the results obtained by BTTR after some steps diverge from the actual values. In this case we

Table 1 Success (“Ok”) and failure (“No”) results obtained by FTTR and BTTR when computing the sequence of orthogonal polynomials associated to the weight ω , considering many different values of α and β

$\omega(x)$	Support	Name	FTTR	BTTR
$(1-x)^\alpha(1+x)^\beta$ $\alpha, \beta > -1,$	$[-1, 1]$	Jacobi	Ok	Ok
$x^\alpha e^{-x}, \alpha > -1,$	$[0, \infty]$	Generalized Laguerre	Ok	No
$ x ^{2\alpha} e^{-x^2}, \alpha > -1/2$	$[-\infty, \infty]$	Generalized Hermite	Ok	No
$x^\alpha \log \frac{1}{x}, \alpha > -1$	$[0, 1]$	algebraic logarithmic	Ok	Ok
$e^{-x}(1+e^{-x})^2$	$[-\infty, \infty]$	logistic	Ok	No
$\frac{1}{\sqrt{(1-\alpha t^2)(1-t^2)}}, 0 < \alpha < 1,$	$[-1, 1]$	elliptic	Ok	Ok
		Hahn	No	Ok

report an “Ok” in column FTTR and a “No” in column BTTR in Table 1. In Fig. 1 (right) we show the results for the Hahn polynomials of degree $j = 0, 1, 2, \dots, 127$, evaluated in the fourth smallest zero of the Hahn polynomial of degree 128, computed in double precision with FTTR and BTTR, for $\alpha = \beta = -.5$. In this case, FTTR does not compute the sequence of Hahn polynomials in an accurate way, while BTTR does. We thus report a “No” in column FTTR and an “Ok” in column BTTR in Table 1.

The Hahn polynomials in Table 1 are discrete orthogonal polynomials on the discrete set of n points $\{0, 1, 2, \dots, n - 1\}$ with respect to the weight $\binom{\alpha+k}{k} \binom{\beta+n-1-k}{n-1-k}$, $\alpha, \beta > -1, k = 0, 1, 2, \dots, n - 1$.

Hence, since experimentally it is not clear which of FTTR and BTTR can be chosen, we describe an algorithm, called LMV, that combines the FTTR and the BTTR methods in order to compute the sequence (10) with high relative accuracy.

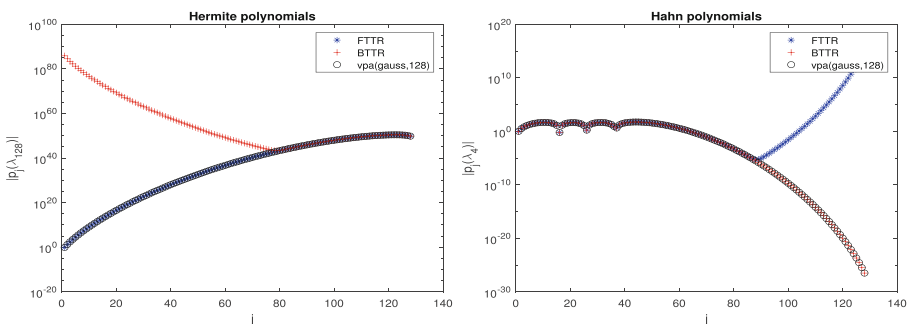


Fig. 1 Left: plot of the absolute values of the Hermite polynomials of degree $j, j = 0, 1, \dots, 127$, evaluated in the largest zero of the Hermite polynomial of degree 128, and computed by FTTR in double precision (denoted by “*”), by BTTR in double precision (denoted by “+”) and by `gauss` with extended precision (128 digits) (denoted by “o”). Right: plot of the absolute values of the Hahn polynomials of degree $j, j = 0, 1, \dots, 127$, evaluated in the fourth smallest zero of the Hahn polynomial of degree 128, and computed by FTTR in double precision (denoted by “*”), by BTTR in double precision (denoted by “o”) and by `gauss` with extended precision (128 digits) (denoted by “o”)

Let $\bar{\lambda}$ be a zero of $p_n(x)$ and let us denote by $\tilde{\mathbf{p}}$ and by $\hat{\mathbf{p}}$ the following vectors, computed by FTTR and BTTR, respectively:

$$\tilde{\mathbf{p}} = \frac{1}{\sqrt{\mu_0}} \frac{1}{\tilde{p}_0(\bar{\lambda})} \begin{bmatrix} \tilde{p}_0(\bar{\lambda}) \\ \tilde{p}_1(\bar{\lambda}) \\ \vdots \\ \tilde{p}_{n-2}(\bar{\lambda}) \\ \tilde{p}_{n-1}(\bar{\lambda}) \end{bmatrix}, \quad \hat{\mathbf{p}} = \frac{1}{\sqrt{\mu_0}} \frac{1}{\hat{p}_0(\bar{\lambda})} \begin{bmatrix} \hat{p}_0(\bar{\lambda}) \\ \hat{p}_1(\bar{\lambda}) \\ \vdots \\ \hat{p}_{n-2}(\bar{\lambda}) \\ \hat{p}_{n-1}(\bar{\lambda}) \end{bmatrix}.$$

The following theorem emphasizes the relationship between the sequence $\tilde{p}_j(\bar{\lambda})$, $j = 1, \dots, n - 1$, computed by (1) and the columns of the \tilde{Q} factor of the QR factorization of $J - \bar{\lambda}I_n$.

Theorem 2 *Let $\bar{\lambda}$ be a zero of $p_n(x)$, and let $\tilde{G}_1, \tilde{G}_2, \dots, \tilde{G}_{n-1}$ be the sequence of Givens rotations*

$$\tilde{G}_i = \begin{bmatrix} I_{i-1} & & & & \\ & \tilde{c}_i & \tilde{s}_i & & \\ & -\tilde{s}_i & \tilde{c}_i & & \\ & & & I_{n-i-1} & \\ & & & & \end{bmatrix}, \quad i = 1, \dots, n - 1, \tag{11}$$

such that

$$\tilde{G}_{n-1} \tilde{G}_{n-2} \cdots \tilde{G}_2 \tilde{G}_1 (J - \bar{\lambda}I_n) = R,$$

with $R \in \mathbb{R}^{n \times n}$ upper triangular. Let $\tilde{p}_0(\bar{\lambda}), \tilde{p}_1(\bar{\lambda}), \dots, \tilde{p}_{n-2}(\bar{\lambda}), \tilde{p}_{n-1}(\bar{\lambda})$ be the sequence of orthogonal polynomials evaluated in $\bar{\lambda}$ by means of the three-term recurrence relation (1). Then

$$\tilde{G}_1^T \tilde{G}_2^T \cdots \tilde{G}_i^T \mathbf{e}_{i+1} = \tilde{v}_i \begin{bmatrix} \tilde{p}_0(\bar{\lambda}) \\ \tilde{p}_1(\bar{\lambda}) \\ \vdots \\ \tilde{p}_i(\bar{\lambda}) \\ \mathbf{0}_{n-i-1} \end{bmatrix}, \quad i = 1, \dots, n - 1, \tag{12}$$

with $\tilde{v}_i \in \mathbb{R}, \tilde{v}_i \neq 0$.

Proof We prove (12) by induction. Let $\tilde{p}_0(\bar{\lambda}) = 1/\sqrt{\mu_0}$. Then, by (1),

$$\tilde{p}_1(\bar{\lambda}) = \frac{\bar{\lambda} \tilde{p}_0(\bar{\lambda})}{\gamma_1} = \frac{1}{\sqrt{\mu_0}} \frac{\bar{\lambda}}{\gamma_1}.$$

On the other hand,

$$\tilde{c}_1 = \frac{-\bar{\lambda}}{\sqrt{\bar{\lambda}^2 + \gamma_1^2}}, \quad \tilde{s}_1 = \frac{\gamma_1}{\sqrt{\bar{\lambda}^2 + \gamma_1^2}}.$$

Therefore,

$$\tilde{G}_1^T \mathbf{e}_2 = \begin{bmatrix} -\tilde{s}_1 \\ \tilde{c}_1 \\ \mathbf{0}_{n-2} \end{bmatrix} = \frac{1}{\sqrt{\bar{\lambda}^2 + \gamma_1^2}} \begin{bmatrix} -\gamma_1 \\ -\bar{\lambda} \\ \mathbf{0}_{n-2} \end{bmatrix} = \tilde{v}_1 \begin{bmatrix} \tilde{p}_0(\bar{\lambda}) \\ \tilde{p}_1(\bar{\lambda}) \\ \mathbf{0}_{n-2} \end{bmatrix},$$

with

$$\tilde{v}_1 = -\frac{\gamma_1\sqrt{\mu_0}}{\sqrt{\bar{\lambda}^2 + \gamma_1^2}}.$$

Let us suppose that (12) holds for i , $1 \leq i < n$ and let us prove (12) for $i + 1$.

Observe that

$$\tilde{G}_1 \cdots \tilde{G}_2 \tilde{G}_1 (J - \bar{\lambda} I_n) = \begin{bmatrix} \times & \times & \times & & & & & & \\ & \ddots & \ddots & \ddots & & & & & \\ & & \times & \times & & \times & & & \\ & & & v_i & & \tilde{c}_i \gamma_{i+1} & & & \\ & & & \gamma_{i+1} & \theta_{i+2} - \bar{\lambda} & \ddots & & & \\ & & & & & \ddots & \ddots & & \gamma_{n-1} \\ & & & & & & \gamma_{n-1} & \theta_n - \bar{\lambda} & \end{bmatrix}, \tag{13}$$

with $v_i = -\tilde{s}_i \tilde{c}_{i-1} \gamma_i + \tilde{c}_i (\theta_{i+1} \bar{\lambda})$. Moreover, by the induction hypothesis (12),

$$\tilde{G}_1^T \cdots \tilde{G}_{i-1}^T \tilde{G}_i^T \mathbf{e}_{i+1} = \begin{bmatrix} (-1)^i \prod_{j=1}^i \tilde{s}_j \\ (-1)^{i-1} \tilde{c}_1 \prod_{j=2}^i \tilde{s}_j \\ (-1)^{i-2} \tilde{c}_2 \prod_{j=3}^i \tilde{s}_j \\ \vdots \\ \tilde{c}_{i-2} \tilde{s}_{i-1} \tilde{s}_i \\ -\tilde{c}_{i-1} \tilde{s}_i \\ \tilde{c}_i \\ \mathbf{0}_{n-i-1} \end{bmatrix} = \tilde{v}_i \begin{bmatrix} \tilde{p}_0(\bar{\lambda}) \\ \tilde{p}_1(\bar{\lambda}) \\ \vdots \\ \tilde{p}_i(\bar{\lambda}) \\ \mathbf{0}_{n-i-1} \end{bmatrix}, \quad i = 1, \dots, n-1.$$

By (1) and the induction hypothesis (12),

$$\tilde{p}_{i+1}(\bar{\lambda}) = \frac{\bar{\lambda} \tilde{p}_i(\bar{\lambda}) - \gamma_i \tilde{p}_{i-1}(\bar{\lambda})}{\gamma_{i+1}} = \frac{\bar{\lambda} \tilde{c}_i + \gamma_i \tilde{c}_{i-1} \tilde{s}_i}{\tilde{v}_i \gamma_{i+1}}.$$

On the other hand, from (13),

$$\tilde{c}_{i+1} = \frac{v_i}{\xi_{i+1}} = \frac{-\tilde{s}_i \tilde{c}_{i-1} \gamma_i - \tilde{c}_i \bar{\lambda}}{\xi_{i+1}} \quad \text{and} \quad \tilde{s}_{i+1} = \frac{\gamma_{i+1}}{\xi_{i+1}},$$

with $\xi_{i+1} = \sqrt{(-\tilde{s}_i \tilde{c}_{i-1} \gamma_i - \tilde{c}_i \bar{\lambda})^2 + \gamma_{i+1}^2}$. Hence,

$$\tilde{p}_{i+1}(\bar{\lambda}) = -\frac{1}{\tilde{v}_i} \frac{\tilde{c}_{i+1}}{\tilde{s}_{i+1}}.$$

Therefore,

$$\begin{aligned}
 \tilde{G}_1^T \cdots \tilde{G}_i^T \tilde{G}_{i+1}^T \mathbf{e}_{i+2} &= \begin{bmatrix} (-1)^{i+1} \prod_{j=1}^{i+1} \tilde{s}_j \\ (-1)^i \tilde{c}_1 \prod_{j=2}^{i+1} \tilde{s}_j \\ (-1)^{i-1} \tilde{c}_2 \prod_{j=3}^{i+1} \tilde{s}_j \\ \vdots \\ \tilde{c}_{i-1} \tilde{s}_i \tilde{s}_{i+1} \\ -\tilde{c}_i \tilde{s}_{i+1} \\ \tilde{c}_{i+1} \\ \mathbf{0}_{n-i-2} \end{bmatrix} = -\tilde{s}_{i+1} \begin{bmatrix} (-1)^i \prod_{j=1}^i \tilde{s}_j \\ (-1)^{i-1} \tilde{c}_1 \prod_{j=2}^i \tilde{s}_j \\ (-1)^{i-2} \tilde{c}_2 \prod_{j=3}^i \tilde{s}_j \\ \vdots \\ -\tilde{c}_{i-1} \tilde{s}_i \\ \tilde{c}_i \\ -\frac{\tilde{c}_{i+1}}{\tilde{s}_{i+1}} \\ \mathbf{0}_{n-i-2} \end{bmatrix} \\
 &= \tilde{v}_{i+1} \begin{bmatrix} \tilde{p}_0(\tilde{\lambda}) \\ \tilde{p}_1(\tilde{\lambda}) \\ \vdots \\ \tilde{p}_i(\tilde{\lambda}) \\ \tilde{p}_{i+1}(\tilde{\lambda}) \\ \mathbf{0}_{n-i-2} \end{bmatrix},
 \end{aligned}$$

with $\tilde{v}_{i+1} = -\tilde{s}_{i+1} \tilde{v}_i$. □

Remark 2 The matrix $\tilde{Q} = \tilde{G}_1^T \cdots \tilde{G}_{n-2}^T \tilde{G}_{n-1}^T$ is the orthogonal upper Hessenberg matrix

$$\tilde{Q} = \begin{bmatrix} \tilde{c}_1 - \tilde{s}_1 \tilde{c}_2 & \tilde{s}_1 \tilde{s}_2 \tilde{c}_3 & \cdots & (-1)^{n-2} \tilde{c}_{n-1} \prod_{i=1}^{n-2} \tilde{s}_i & (-1)^{n-1} \prod_{i=1}^{n-1} \tilde{s}_i \\ \tilde{s}_1 & \tilde{c}_1 \tilde{c}_2 & -\tilde{c}_1 \tilde{s}_2 \tilde{c}_3 & \cdots & (-1)^{n-3} \tilde{c}_1 \tilde{c}_{n-1} \prod_{i=2}^{n-2} \tilde{s}_i & (-1)^{n-2} \tilde{c}_1 \prod_{i=2}^{n-1} \tilde{s}_i \\ & \tilde{s}_2 & \tilde{c}_2 \tilde{c}_3 & \cdots & \cdots & \vdots \\ & & \cdots & \cdots & -\tilde{c}_{n-3} \tilde{s}_{n-2} \tilde{c}_{n-1} & \tilde{c}_{n-3} \tilde{s}_{n-2} \tilde{s}_{n-1} \\ & & & \tilde{s}_{n-2} & \tilde{c}_{n-2} \tilde{c}_{n-1} & -\tilde{c}_{n-2} \tilde{s}_{n-1} \\ & & & & \tilde{s}_{n-1} & \tilde{c}_{n-1} \end{bmatrix}.$$

since $J - \tilde{\lambda} I_n$ is a tridiagonal matrix. Therefore, by Theorem 2, the subvector made by the first i entries of $\mathbf{p}(\tilde{\lambda})$ is parallel to the vectors made by the first i entries of the columns j of \tilde{Q} , with $j \geq i$.

Theorem 3 shows the relationship between the sequence $\{\hat{p}_j(\tilde{\lambda})\}_{j=0}^{n-1}$ and the columns of the factor \hat{Q} of the QL factorization of $J - \tilde{\lambda} I_n$. For the sake of brevity, we omit the proof since it is very similar to the one of Theorem 2.

Theorem 3 Let $\bar{\lambda}$ be a zero of $p_n(x)$, and let $\hat{G}_1, \hat{G}_2, \dots, \hat{G}_{n-1}$ be the sequence of Givens rotations

$$\hat{G}_i = \begin{bmatrix} I_{n-i-1} & & & \\ & \hat{c}_i & \hat{s}_i & \\ & -\hat{s}_i & \hat{c}_i & \\ & & & I_{i-1} \end{bmatrix}, \quad i = 1, \dots, n - 1, \tag{14}$$

such that

$$\hat{G}_{n-1}^T \hat{G}_{n-2}^T \dots \hat{G}_2^T \hat{G}_1^T (J - \bar{\lambda}I_n) = L,$$

with $L \in \mathbb{R}^{n \times n}$ lower triangular. Let $\hat{p}_{n-1}(\bar{\lambda}), \hat{p}_{n-2}(\bar{\lambda}), \dots, \hat{p}_1(\bar{\lambda}), \hat{p}_0(\bar{\lambda})$, be the sequence evaluated in $\bar{\lambda}$ by the three-term recurrence relation (1) in a backward fashion, with $\hat{p}_{n-1}(\bar{\lambda})$ fixed. Then

$$\hat{G}_1 \hat{G}_2 \dots \hat{G}_i e_{n-i} = \begin{bmatrix} \mathbf{0}_{n-i-1} \\ \hat{c}_i \\ -\hat{c}_{i-1} \hat{s}_i \\ \hat{c}_{i-2} \hat{s}_{i-1} \hat{s}_i \\ \vdots \\ (-1)^{i-2} \hat{c}_2 \prod_{j=3}^i \hat{s}_j \\ (-1)^{i-1} \hat{c}_1 \prod_{j=2}^i \hat{s}_j \\ (-1)^i \prod_{j=1}^i \hat{s}_j \end{bmatrix} = \hat{v}_i \begin{bmatrix} \mathbf{0}_{n-i-1} \\ \hat{p}_{n-i-1}(\bar{\lambda}) \\ \hat{p}_{n-i-2}(\bar{\lambda}) \\ \vdots \\ \hat{p}_{n-2}(\bar{\lambda}) \\ \hat{p}_{n-1}(\bar{\lambda}) \end{bmatrix}, \quad i = 1, \dots, n-1, \tag{15}$$

with $\hat{v}_i \in \mathbb{R}, \hat{v}_i \neq 0$.

Theorem 2 shows that the vector $\hat{\mathbf{p}}$, computed using FTTR, can also be obtained by applying either one step of the implicit QR algorithm with shift $\bar{\lambda}$ to J or computing the QR factorization of $J - \bar{\lambda}I_n$, since the orthogonal matrices generated by both methods are the same. The last column of the orthogonal matrices generated by both methods will be parallel to $\hat{\mathbf{p}}$. Therefore, *forward instability* can occur in the computation of $\hat{\mathbf{p}}$ if *premature convergence* occurs in one step of the forward implicit QR (FIQR) method with shift $\bar{\lambda}$ to J [17, 21].

On the other hand, Theorem 3 shows that the vector $\hat{\mathbf{p}}$, computed applying BTTR, can also be obtained applying either one step of the backward implicit QR algorithm with shift $\bar{\lambda}$ to J or computing the QL factorization of $J - \bar{\lambda}I_n$. The first column of the orthogonal matrices generated by both methods will be parallel to $\hat{\mathbf{p}}$. Therefore, *forward instability* can occur in the computation of $\hat{\mathbf{p}}$ if *premature convergence* occurs in one step of the backward implicit QL (BIQL) method with shift $\bar{\lambda}$ to J [17, 21].

The premature convergence of the implicit QR method with shift $\bar{\lambda}$ depends on the distance between the eigenvalues of the consecutive matrices $J_{1:i,1:i}$ and $J_{1:i+1,1:i+1}$, $i = 1, 2, \dots, n - 1$. Hence, if $\lambda_j^{(i)}, j = 1, \dots, i$, and $\lambda_j^{(i+1)}, j = 1, \dots, i + 1$, are the eigenvalues of $J_{1:i,1:i}$ and $J_{1:i+1,1:i+1}$, respectively, then, by the Cauchy interlacing Theorem [22],

$$\lambda_1^{(i+1)} < \lambda_1^{(i)} < \lambda_2^{(i+1)} < \lambda_2^{(i)} < \dots < \lambda_i^{(i+1)} < \lambda_i^{(i)} < \lambda_{i+1}^{(i+1)}.$$

If $|\lambda_j^{(i+1)} - \lambda_j^{(i)}|$ and $|\lambda_j^{(i)} - \lambda_{j+1}^{(i+1)}|$, $j = 1, \dots, i$, $i = 1, \dots, n - 1$, are sufficiently large, then premature convergence does not occur in one step of the implicit forward QR algorithm with shift $\bar{\lambda}$ and, hence, FTTR computes the sequence (10) accurately. This is the reason why FTTR computes the sequence (10) accurately for the Chebyshev polynomials $T_j(x)$ of the first kind, i.e., Jacobi polynomials associated with the weight $\frac{1}{\sqrt{1-x^2}}$ in the interval $[-1, 1]$, since the distance of the zeros of two consecutive polynomials $T_j(x)$ and $T_{j+1}(x)$ is of order $\mathcal{O}\left(\frac{1}{j^2}\right)$ at least.

In [17, 18], an algorithm combining one step of FIQR and one step of BIQL with shift $\bar{\lambda}$ is described in order to compute the corresponding eigenvector. Therefore, the eigenvector associated with $\bar{\lambda}$ is computed combining the first $\bar{j} - 1$ Givens rotations of FIQR with shift $\bar{\lambda}$ and the first $n - \bar{j}$ rotations of BIQL with shift $\bar{\lambda}$. It is proven that each eigenvector is computed accurately with $\mathcal{O}(n)$ floating point operations. Once the eigenvector is computed, the weights are obtained by applying (9).

Following [17, 18], we now describe a recursive procedure to determine an interval in which the index \bar{j} lies. Let us consider the sequence of Givens rotations $\tilde{G}_i \in \mathbb{R}^{n \times n}$, $i = 1, \dots, n - 1$, defined in (11). It turns out that

$$\tilde{G}_1 \tilde{G}_2 \cdots \tilde{G}_i \tilde{\mathbf{p}} = \frac{1}{\sqrt{\mu_0}} \frac{1}{\tilde{p}_0(\bar{\lambda})} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{\sum_{\ell=1}^i \tilde{p}_\ell^2(\bar{\lambda})} \\ \tilde{p}_{i+1}(\bar{\lambda}) \\ \vdots \\ \tilde{p}_{n-1}(\bar{\lambda}) \end{bmatrix}.$$

Then we compute the sequence of normalized vectors $\tilde{\mathbf{v}}_i \in \mathbb{R}^i$, $i = 1, \dots, n$, in the following way,

$$\begin{cases} \tilde{\mathbf{v}}_i = 1, & i = 1, \\ \mathbf{w}_i = \tilde{G}_1^T \tilde{G}_2^T \cdots \tilde{G}_{i-1}^T \mathbf{e}_i, \tilde{\mathbf{v}}_i = \mathbf{w}_i(1:i) = \begin{bmatrix} \tilde{s}_{i-1} \tilde{\mathbf{v}}_{i-1} \\ \tilde{c}_{i-1} \end{bmatrix}, & i = 2, \dots, n, \end{cases}$$

and the Rayleigh quotients $\tilde{\lambda}^{(i)} = \tilde{\mathbf{v}}_i^T J_{1:i,1:i} \tilde{\mathbf{v}}_i$, for $i = 2, \dots, n$, as follows,

$$\begin{aligned} \tilde{\lambda}^{(i)} &= \tilde{\mathbf{v}}_i^T J_{1:i,1:i} \tilde{\mathbf{v}}_i \\ &= [\tilde{s}_{i-1} \tilde{\mathbf{v}}_{i-1}^T, \tilde{c}_{i-1}] J_{1:i,1:i} \begin{bmatrix} \tilde{s}_{i-1} \tilde{\mathbf{v}}_{i-1} \\ \tilde{c}_{i-1} \end{bmatrix} \\ &= [\tilde{s}_{i-1} \tilde{\mathbf{v}}_{i-1}^T, \tilde{c}_{i-1}] \left[\begin{array}{c|c} J_{1:i-1,1:i-1} & \gamma_{i-1} \mathbf{e}_{i-1} \\ \hline \gamma_{i-1} \mathbf{e}_{i-1}^T & \theta_i \end{array} \right] \begin{bmatrix} \tilde{s}_{i-1} \tilde{\mathbf{v}}_{i-1} \\ \tilde{c}_{i-1} \end{bmatrix} \\ &= \tilde{s}_{i-1}^2 \tilde{\mathbf{v}}_{i-1}^T J_{1:i-1,1:i-1} \tilde{\mathbf{v}}_{i-1} + 2\tilde{s}_{i-1} \tilde{c}_{i-1} \gamma_{i-1} + \tilde{c}_{i-1}^2 \theta_i \\ &= \tilde{s}_{i-1}^2 \tilde{\lambda}^{(i-1)} + 2\tilde{s}_{i-1} \tilde{c}_{i-1} \tilde{c}_{i-2} \gamma_{i-1} + \tilde{c}_{i-1}^2 \theta_i, \end{aligned}$$

with $\tilde{c}_0 = 1$, $\tilde{\lambda}^{(1)} = \theta_1$.

If premature convergence occurs at step \tilde{j} of FIQR with shift $\bar{\lambda}$, then $\tilde{\mathbf{v}}_{\tilde{j}}$ is the eigenvector of $J_{1:\tilde{j}}$ associated with the eigenvalue $\bar{\lambda} \approx \tilde{\lambda}^{(\tilde{j})} = \tilde{\mathbf{v}}_{\tilde{j}}^T J_{1:\tilde{j},1:\tilde{j}} \tilde{\mathbf{v}}_{\tilde{j}}$. It then follows that $\bar{j} \leq \tilde{j}$ (see [17, 18]).

Similarly, let $\hat{G}_i \in \mathbb{R}^{n \times n}$, $i = 1, \dots, n$ be the sequence of Givens rotations defined in (11). Then

$$\hat{G}_1^T \hat{G}_2^T \cdots \hat{G}_i^T = \frac{1}{\sqrt{\mu_0}} \frac{1}{\hat{p}_0(\bar{\lambda})} \begin{bmatrix} \hat{p}_0(\bar{\lambda}) \\ \vdots \\ \frac{\hat{p}_{n-i-2}(\bar{\lambda})}{\sqrt{\sum_{\ell=1}^{i+1} \hat{p}_{n-i}^2(\bar{\lambda})}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad i = 1, \dots, n - 1.$$

We construct the sequence of vectors $\hat{\mathbf{v}}_i \in \mathbb{R}^i$, $i = 1, \dots, n$, as follows,

$$\hat{\mathbf{v}}_i = \hat{\mathbf{w}}_i(n - i + 1 : n) = \begin{bmatrix} \hat{c}_{i-1} \\ \hat{s}_{i-1} \hat{\mathbf{v}}_{i-1} \end{bmatrix}, \quad i = 2, \dots, n,$$

where $\hat{\mathbf{w}}_i = \hat{G}_1 \hat{G}_2 \cdots \hat{G}_{i-1} \mathbf{e}_{n-i+1}$, and $\hat{\mathbf{v}}_1 \equiv 1$, and the Rayleigh quotients $\hat{\lambda}^{(i)} = \hat{\mathbf{v}}_i^T J_{n-i+1:n,n-i+1:n} \hat{\mathbf{v}}_i$, as,

$$\begin{aligned} \hat{\lambda}^{(i)} &= \hat{\mathbf{v}}_i^T J_{n-i+1:n,n-i+1:n} \hat{\mathbf{v}}_i \\ &= [\hat{c}_{i-1}, \hat{s}_{i-1} \hat{\mathbf{v}}_{i-1}^T] J_{n-i+1:n,n-i+1:n} \begin{bmatrix} \hat{c}_{i-1} \\ \hat{s}_{i-1} \hat{\mathbf{v}}_{i-1} \end{bmatrix} \\ &= [\hat{c}_{i-1}, \hat{s}_{i-1} \hat{\mathbf{v}}_{i-1}^T] \left[\begin{array}{c|c} \theta_{n-i+1} & \gamma_{n-i+1} \mathbf{e}_{i-1}^T \\ \hline \gamma_{n-i+1} \mathbf{e}_{i-1} & J_{n-i+2:n,n-i+2:n} \end{array} \right] \begin{bmatrix} \hat{c}_{i-1} \\ \hat{s}_{i-1} \hat{\mathbf{v}}_{i-1} \end{bmatrix} \\ &= \hat{s}_{i-1}^2 \hat{\mathbf{v}}_{i-1}^T J_{n-i+2:n,n-i+2:n} \hat{\mathbf{v}}_{i-1} + 2\hat{s}_{i-1} \hat{c}_{i-1} \hat{c}_{i-1} \gamma_{n-i+1} + \hat{c}_{i-1}^2 \theta_{n-i+1} \\ &= \hat{s}_{i-1}^2 \hat{\lambda}^{(i-1)} + 2\hat{s}_{i-1} \hat{c}_{i-1} \hat{c}_{i-2} \gamma_{n-i+1} + \hat{c}_{i-1}^2 \theta_{n-i+1}, \end{aligned}$$

with $\hat{\lambda}^{(1)} = \theta_n$ and $\hat{c}_0 = 1$.

It turns out that if premature convergence occurs at step \hat{j} , $1 \leq \hat{j} \leq n - 1$, in IBIQL with shift $\bar{\lambda}$, then $\hat{\mathbf{v}}_{\hat{j}}$ is the eigenvector of $J_{n-\hat{j}:n,n-\hat{j}:n}$ associated with the eigenvalue $\bar{\lambda} \approx \tilde{\lambda}^{(\hat{j})} = \tilde{\mathbf{v}}_{\hat{j}}^T J_{1:\hat{j}} \tilde{\mathbf{v}}_{\hat{j}}$.

Since J is an irreducible tridiagonal matrix, it was shown in [17, 18, 21] that if premature convergence occurs at step \tilde{j} of FIQR with shift $\bar{\lambda}$, with $1 \leq \hat{j} \leq n - 1$, then premature convergence can only occur at the step \hat{j} of BIQL with shift $\bar{\lambda}$, with $1 \leq \hat{j} \leq \tilde{j}$.

This suggests the Algorithm 1, written in a Matlab-style, to compute the interval $[\hat{j}, \tilde{j}]$ in which \bar{j} lies.

Algorithm 1 Given as input the vectors $\tilde{\mathbf{p}}, \hat{\mathbf{p}}$, and the root $\bar{\lambda}$ of $p_n(x)$, the function `find_j` computes the interval $[\hat{j}, \tilde{j}]$, $1 \leq \hat{j} \leq \bar{j} \leq \tilde{j} \leq n$.

```

function[ $\hat{j}, \tilde{j}$ ]= find_j( $\tilde{\mathbf{p}}_n, \hat{\mathbf{p}}_n, \bar{\lambda}, n, tol$ )
 $i = 1; \lambda_0^{(F)} = \theta_1; \tilde{c}^{(F)} = 1; tF = 0; \tilde{j} = 1; \hat{j} = n;$ 
while  $tF \leq 3$ ,
     $G = \text{givens}(\tilde{p}_{i+1}, \tilde{p}_i); \quad \tilde{c}_i = G(1, 1); \tilde{s}_i = G(1, 2);$ 
     $\tilde{\mathbf{p}}_{i:i+1} = G^T \tilde{\mathbf{p}}_{i:i+1};$ 
     $\lambda_1^{(F)} = \tilde{s}_i^2 \lambda_0^{(F)} + 2\gamma_i \tilde{s}_i \tilde{c}_i \tilde{c}^{(F)} + \tilde{c}_i^2 \theta_{i+1};$ 
    if  $|\lambda_1^{(F)} - \bar{\lambda}| \leq tol,$ 
         $tF = tF + 1; \quad \tilde{j} = i + 1;$ 
    else
         $tF = 0;$ 
    end
     $i = i + 1;$ 
    if  $i == n \ \& \ tF == 1,$ 
         $tF = 4;$ 
    end
     $\lambda_0^{(F)} = \lambda_1^{(F)}; \tilde{c}^{(F)} = \tilde{c}_i;$ 
end
if  $tF < 4,$ 
     $\lambda_0^{(B)} = \theta_n; \hat{c}^{(B)} = 1; \quad i = n - 1; tB = 0;$ 
    while  $tB < 3,$ 
         $G = \text{givens}(\hat{p}_i, \hat{p}_{i+1}); \quad \hat{c}_i = G(1, 1); \hat{s}_i = G(1, 2);$ 
         $\hat{\mathbf{p}}_{i:i+1} = G \hat{\mathbf{p}}_{i:i+1};$ 
         $\lambda_1^{(B)} = \hat{s}_i^2 \lambda_0^{(F)} + 2\gamma_i \hat{s}_i \hat{c}_i \hat{c}^{(B)} + \hat{c}_i^2 \theta_i;$ 
        if  $|\lambda_1^{(B)} - \bar{\lambda}| \leq tol,$ 
             $tB = tB + 1; \hat{j} = i;$ 
        else
             $tB = 0;$ 
            if  $tB == 1 \ \& \ i == 1,$ 
                 $tB = 4;$ 
            end
             $i = i - 1; \quad \lambda_0^{(B)} = \lambda_1^{(B)}; \hat{c}^{(B)} = \hat{c}_i;$ 
        end
    end
end
end

```

Once the interval $[\hat{j}, \tilde{j}]$ is determined, the index \bar{j} is chosen as the index with the maximum element in absolute value in the subvector $\tilde{\mathbf{p}}_{\hat{j}:\tilde{j}}$ [17, 18].

Remark 3 Given $\bar{\lambda}$, a similar recursion holds for estimating if premature convergence occurs in a singular value of the bidiagonal matrix B .

Remark 4 We have described in Section 4 that the sequence $p_\ell(\lambda_i)$, $\ell = 0, 1, \dots, n - 1$, can be retrieved from the eigenvector of J associated with λ_i . Since

the eigenvalue decomposition of J can be obtained from the singular value decomposition of B , then, the eigenvector sequences can be retrieved in a similar way from the left and right singular vectors associated with the singular value λ_i of the corresponding bidiagonal matrix. For the sake of brevity, we omit the details.

6 Stability of the eigenvectors and weights

In this section we analyze the sensitivity of the calculation of the eigenvector of the tridiagonal matrix J and the sensitivity of the corresponding weight of the GQR, for a particular eigenvalue λ_j . Let us define the shifted matrix as $T := J - \lambda_j I_n$, where T is tridiagonal and unreduced, let \mathbf{p} be the corresponding vector of orthogonal polynomials evaluated at λ_j , i.e., $T\mathbf{p} = 0$, and let ω be the corresponding weight, i.e., $\omega := 1/(\mathbf{p}^T \mathbf{p})$. Let us now consider any nonzero element $p_i \neq 0$ of \mathbf{p} . We then partition the rows of the matrix T as follows:

$$T = \begin{bmatrix} T_{1:i-1,:} \\ \mathbf{t}_i \\ T_{i+1:n,:} \end{bmatrix},$$

implying that $T_{1:0}$ and $T_{n+1:n,:}$ are void matrices. Then the equation $\mathbf{p}^T T = 0$ and the fact that $p_i \neq 0$, implies that the row vector \mathbf{t}_i is in the row space of the remaining rows of T . If we then construct the matrix

$$T_{(i)} := \begin{bmatrix} T_{1:i-1,:} \\ 0 \\ T_{i+1:n,:} \end{bmatrix} = (I_n - \mathbf{e}_i \mathbf{e}_i^T) T.$$

then the two systems of equations $T\mathbf{p} = 0$ and $T_{(i)}\mathbf{p} = 0$ have the same one dimensional set of solutions, i.e., their kernels are the same:

$$\ker T = \ker T_{(i)} = \text{Im } \mathbf{p}.$$

We now compare their sensitivities with respect to perturbations of the data. Let us denote the normalized vector $\mathbf{p}/\|\mathbf{p}\|_2$ by \mathbf{q} , which is thus the normalized eigenvector of J . It follows that $q_i \neq 0$ and

$$M \begin{bmatrix} T_{1:i-1,:} \\ \mathbf{t}_i \\ T_{i+1:n,:} \end{bmatrix} = \begin{bmatrix} T_{1:i-1,:} \\ 0 \\ T_{i+1:n,:} \end{bmatrix}, \text{ with } \begin{bmatrix} \mathbf{q}_{1:i-1} \\ q_i \\ \mathbf{q}_{i+1:n} \end{bmatrix} := \mathbf{q}, \quad M := \begin{bmatrix} I_{i-1} & 0 & 0 \\ \mathbf{q}_{1:i-1}^T & q_i & \mathbf{q}_{i+1:n}^T \\ 0 & 0 & I_{n-1} \end{bmatrix}.$$

This then yields the following theorem.

Theorem 4 *Let*

$$T_{(i)} := (I_n - \mathbf{e}_i \mathbf{e}_i^T) T, \quad \text{and} \quad \hat{\mathbf{q}} := (I_n - \mathbf{e}_i \mathbf{e}_i^T) \mathbf{q} = \begin{bmatrix} \mathbf{q}_{1:i-1} \\ 0 \\ \mathbf{q}_{i+1:n} \end{bmatrix},$$

then for any $q_i \neq 0$, we have

$$\sigma_{n-1}(T) |q_i| / \sqrt{2} < \sigma_{n-1}(T_{(i)}) \leq \sigma_{n-1}(T) \tag{16}$$

and for any $|q_i| = \|\mathbf{q}\|_\infty$, or, equivalently, $|p_i| = \|\mathbf{p}\|_\infty$, we have

$$\sigma_{n-1}(T)/\sqrt{2n} < \sigma_{n-1}(T_{(i)}) \leq \sigma_{n-1}(T). \tag{17}$$

Proof The Cauchy inequalities for singular values yields $\sigma_{n-1}(T_{(i)}) \leq \sigma_{n-1}(T)$ since we deleted one row of the matrix T to obtain $T_{(i)}$. Let $\mathbf{Q} \in \mathbb{R}^{n \times (n-1)}$ be the orthogonal complement of \mathbf{q} , i.e., $\mathbf{Q}^T \mathbf{Q} = I_{n-1}$ and $\mathbf{Q}^T \mathbf{q} = 0$. Then

$$M\mathbf{T}\mathbf{Q} = T_{(i)}\mathbf{Q}, \quad \sigma_{n-1}(T) = \sigma_{\min}(T\mathbf{Q}), \quad \sigma_{n-1}(T_{(i)}) = \sigma_{\min}(T_{(i)}\mathbf{Q}).$$

Let $\mathbf{v} \in \mathbb{R}^{n-1}$ be such that $\|\mathbf{v}\|_2 = 1$, and $\|M\mathbf{T}\mathbf{Q}\mathbf{v}\|_2 = \sigma_{\min}(M\mathbf{T}\mathbf{Q}) = \sigma_{\min}(T_{(i)}\mathbf{Q})$. We then have $\|M\mathbf{T}\mathbf{Q}\mathbf{v}\|_2 \geq \sigma_{\min}(M)\|T\mathbf{Q}\mathbf{v}\|_2 \geq \sigma_{\min}(M)\sigma_{\min}(T\mathbf{Q})$, which implies

$$\sigma_{\min}(M) \cdot \sigma_{\min}(T\mathbf{Q}) \leq \sigma_{\min}(T_{(i)}\mathbf{Q}).$$

Moreover,

$$M\mathbf{M}^T = I + \mathbf{e}_i \hat{\mathbf{q}}^T + \hat{\mathbf{q}} \mathbf{e}_i^T,$$

and it has been shown in [2, Thm 3.6] that

$$\sigma_{\min}(M) = \sqrt{1 - \|\hat{\mathbf{q}}\|_2} = |q_i| / \sqrt{1 + \|\hat{\mathbf{q}}\|_2} > |q_i| / \sqrt{2}.$$

Putting this together yields (16). The inequalities (17) for $|q_i| = \|\mathbf{q}\|_\infty$ then follow from $|q_i| \geq 1/\sqrt{n}$. □

Let us denote by \mathbf{p}_{fl} the approximation of the vector \mathbf{p} computed by LMV, and let us look at how \mathbf{p}_{fl} is constructed from the matrix equation $T_{(i)}\mathbf{p} = 0$ and show that it is backward stable in the sense that there exists a perturbation Δ such that the computed vector \mathbf{p}_{fl} satisfies exactly the equation

$$(T_{(i)} + \Delta)\mathbf{p}_{fl} = 0, \quad \|\Delta\|_2 \leq \mathcal{O}(\epsilon_M)\|T_{(i)}\|_2, \tag{18}$$

where ϵ_M is the machine precision of the computer used. Once the index i has been chosen, the LMV method computes the vector shared by the kernels of $T_{1:i-1, \cdot}$ and $T_{i+1:n, \cdot}$. Basis vectors for these kernels are respectively given by

$$\alpha \begin{bmatrix} \tilde{\mathbf{p}}_{1:i-1} \\ \tilde{p}_i \\ \tilde{\mathbf{x}} \end{bmatrix} \quad \text{and} \quad \beta \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{p}_i \\ \hat{\mathbf{p}}_{i+1:n} \end{bmatrix},$$

where $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$ are arbitrary. In order to construct a common vector in the two kernels, we impose $\beta = \alpha \tilde{p}_i / \hat{p}_i$, and then choose α such that the common vector corresponds to the initialization $\tilde{p}_0(\lambda_j) = 1/\sqrt{\mu_0}$. The subvectors

$$\alpha \begin{bmatrix} \tilde{\mathbf{p}}_{1:i-1} \\ \tilde{p}_i \end{bmatrix} \quad \text{and} \quad \beta \begin{bmatrix} \hat{p}_i \\ \hat{\mathbf{p}}_{i+1:n} \end{bmatrix},$$

are computed by a forward and backward recurrence using the three-term recurrence of the tridiagonal matrix T . After fixing the starting values in these recurrences, they each can be interpreted as a back substitution of a triangular system of equations. This was analyzed in depth in [16, Ch. 8], from which it follows that the computed

vectors satisfy exactly

$$(\tilde{T}_{1:i-1,:} + \tilde{\Delta}) \begin{bmatrix} \alpha \tilde{\mathbf{p}}_{1:i-1} \\ \alpha \tilde{p}_i \\ 0 \end{bmatrix} = 0, \quad \text{and} \quad (\hat{T}_{i+1:n,:} + \hat{\Delta}) \begin{bmatrix} 0 \\ \beta \hat{p}_i \\ \beta \hat{\mathbf{p}}_{i+1:n} \end{bmatrix} = 0,$$

where $\zeta_n := nu/(1 - nu)^1$ and $\tilde{\Delta}$ and $\hat{\Delta}$ satisfy the elementwise bounds

$$|\tilde{\Delta}| \leq \zeta_2 |\tilde{T}_{1:i-1,:}| \quad \text{and} \quad |\hat{\Delta}| \leq \zeta_2 |\hat{T}_{i+1:n,:}|.$$

Moreover, the above bounds are independent of the scaling factors α and β . Putting this together shows that the proposed method constructs the *computed* vector \mathbf{p}_{fl} that satisfies *exactly* the perturbed system of equations

$$(T_{(i)} + \Delta) \mathbf{p}_{fl} = 0, \quad \text{where} \quad \mathbf{p}_{fl} := \begin{bmatrix} \tilde{\mathbf{p}}_{1:i-1} \\ \tilde{p}_i \\ \hat{\mathbf{p}}_{i+1:n} \end{bmatrix}, \quad \Delta := \begin{bmatrix} \tilde{\Delta} \\ 0 \\ \hat{\Delta} \end{bmatrix}, \quad |\Delta| \leq \zeta_2 |T_{(i)}|. \tag{19}$$

This finally leads to the following theorem.

Theorem 5 *The computed quantities \mathbf{p}_{fl} and $\omega_{fl} := fl(1/\mathbf{p}_{fl}^T \mathbf{p}_{fl})$, obtained by the LMV algorithm with $|p_i| = \|\mathbf{p}\|_\infty$, satisfy the bounds*

$$\|\mathbf{p}_{fl} - \mathbf{p}\|_2 / \|\mathbf{p}_{fl}\|_2 \leq \mathcal{O}(\epsilon_M) \frac{\max_i |\lambda_i|}{\min_{i,i \neq j} (|\lambda_i - \lambda_j|)},$$

and

$$|\omega_{fl} - \omega| / |\omega| \leq \mathcal{O}(\epsilon_M) \frac{\max_i |\lambda_i|}{\min_{i,i \neq j} (|\lambda_i - \lambda_j|)},$$

which implies normwise forward stability for \mathbf{p} and forward stability for the corresponding weight ω .

Proof It follows from the compatible equations $T_{(i)}\mathbf{p} = 0$ and $(T_{(i)} + \Delta)\mathbf{p}_{fl} = 0$ that

$$\|\mathbf{p}_{fl} - \mathbf{p}\|_2 \leq \|\Delta\|_2 \|\mathbf{p}_{fl}\|_2 / \sigma_{n-1}(T_{(i)}).$$

Since Δ is tridiagonal and elementwise bounded by $\zeta_2 \|T\|_2$ it follows that $\|\Delta\|_2 \leq 3\zeta_2 \|T\|_2$. Using this and the bound (17) then implies that

$$\|\mathbf{p} - \mathbf{p}_{fl}\|_2 / \|\mathbf{p}_{fl}\|_2 \leq 3\zeta_2 \sqrt{2n} \|T\|_2 / \sigma_{n-1}(T) = \mathcal{O}(\epsilon_M) \frac{\max_i |\lambda_i|}{\min_{i,i \neq j} (|\lambda_i - \lambda_j|)}. \tag{20}$$

For bounding the relative error in ω , we make use of the relative perturbation theory of norms and inverses, as developed in [16, Chap 3]:

$$|fl(\mathbf{p}^T \mathbf{p}) - \mathbf{p}^T \mathbf{p}| < \zeta_n |\mathbf{p}^T \mathbf{p}|, \quad |fl(1/a) - 1/a| < \zeta_1 |1/a|,$$

which shows that both functions are forward stable in a relative sense. Combining this with the bound (20) yields a similar bound for ω . □

¹The quantity ζ_n is of the order of the unit round-off $u := \frac{1}{2}\epsilon_M$ and corresponds to n rounding errors of a single scalar operation, and follows from the fact that the triangular systems are banded.

We point out that the stability result for the eigenvector is similar to what one has for the eigendecomposition of the matrix J , since it is inversely proportional to the smallest nonzero singular value of T , i.e., to the smallest gap between λ_j and the remaining eigenvalues. The sensitivity of each weight has this same inverse factor, but it has, except for this factor, a forward error that is stable in a relative sense. This is a strong property that is not shared by the eigenvalue method.

7 Numerical examples

In this section we compare the computation of the nodes and weights of n -point GQRs obtained by the proposed method, called LMV, to `gauss`, the `Matlab` function available in [9] and to the methods proposed in [19]. All the experiments were performed in `Matlab ver. R2020b`. In [19], the authors show that the positive nodes and the corresponding weights of a GQR associated with a symmetric weight function can be retrieved from the eigendecomposition of a tridiagonal matrix $J_{\lfloor \frac{n}{2} \rfloor}$ of order $\lfloor \frac{n}{2} \rfloor$, providing the `Matlab` functions displayed in Table 2.

In the first and second example these methods are used to compute n -point GQRs corresponding to the Chebyshev weights of first and second kind, respectively, since their nodes and weights are known. In the third example the proposed method is compared to the function `gauss` in computing an integral on the whole real line.

Example 1 The nodes of the n -point GQR associated with the Chebyshev weight of the first kind,

$$w(x) = \frac{1}{\sqrt{1-x^2}},$$

are $x_j = \cos(\frac{(2j-1)\pi}{2n})$, $j = 1, \dots, n$, the zeros of $\mathcal{T}_n(x)$, the Chebyshev polynomial of the first kind of degree n . Moreover, the weights are $w_j = \pi/n$, $j = 1, \dots, n$. The maxima of the relative errors of the nodes computed by the considered numerical methods

$$\max_j \frac{|\lambda_j - x_j|}{|x_j|}$$

Table 2 Numerical methods for computing the nodes and the weights of GQRs associated with symmetric weight functions proposed in [19]

SymmAG	<code>gauss</code> applied to $J_{\lfloor \frac{n}{2} \rfloor}$
SymmAGw	Optimized Golub and Welsch algorithm applied to $J_{\lfloor \frac{n}{2} \rfloor}$
SymmAw	<code>Matlab's eig</code> for the eigenvalues of $J_{\lfloor \frac{n}{2} \rfloor}$ and the three-term recurrence of the polynomials for the weights
SymmMGw	Optimized Golub and Welsch algorithm applied to the reduced matrix $J_{\lfloor \frac{n}{2} \rfloor}$
SymmMw	<code>eig</code> for the eigenvalues $J_{\lfloor \frac{n}{2} \rfloor}$ and the three-term recurrence of the polynomials for the weights

are reported in Table 3, while the maxima of the relative errors of the computed nodes

$$\max_j \frac{|w_j - \omega_j|}{|w_j|}$$

are reported in Table 4.

In all the cases, the relative errors of the nodes and weights computed by the proposed method are comparable to those of the results yielded by the algorithms proposed in [19].

Example 2 The nodes of the n -point GQR associated with the Chebyshev weight of the second kind,

$$w(x) = \sqrt{1 - x^2},$$

are $x_j = \cos(\frac{\pi}{n+1})$, $j = 1, \dots, n$, the zeros of $\mathcal{U}_n(x)$, the Chebyshev polynomial of the second kind of degree n . Moreover, the weights are $w_j = (1 - x_j^2)^{\frac{\pi}{n+1}}$, $j = 1, \dots, n$. The maxima of the relative errors of the nodes computed by the considered numerical methods are reported in Table 5 while the maxima of the relative errors of the computed nodes are reported in Table 6.

In all the cases, the relative errors of the nodes and weights computed by the proposed method are smaller than those of the results yielded by the algorithms proposed in [19].

Example 3 In this example we consider a GQR for integrals on the whole real line with the Hermite weight $\omega(x) = e^{-x^2}$

$$\int_{-\infty}^{\infty} \omega(x) f(x) dx = \sum_{j=1}^n \omega_j f(\lambda_j) + E_n(f).$$

Table 3 Maxima of the relative errors of the computed nodes for n -point GQRs associated with the Chebyshev weight of the first kind $w(x) = (1 - x^2)^{-1/2}$. For each n , the smallest error is displayed in boldface

	$\max_j \frac{ \lambda_j - x_j }{ x_j }$						
n	LMV	SymmAG	SymmAGWo	SymmAw	SymmMGWo	SymmMw	gauss
32	1.97e-15	6.64e-15	1.03e-14	1.55e-15	1.03e-14	1.55e-15	8.63e-16
64	8.11e-16	2.58e-14	2.20e-14	6.07e-15	2.20e-14	6.07e-15	7.91e-15
128	4.14e-15	2.04e-14	5.95e-14	1.72e-13	5.95e-14	1.72e-13	1.51e-14
256	6.21e-15	3.59e-13	4.39e-13	1.89e-13	4.39e-13	1.89e-13	2.64e-14
512	5.27e-14	1.97e-12	3.71e-12	1.17e-12	3.71e-12	1.17e-12	1.62e-14
1024	2.26e-14	9.26e-12	1.98e-12	3.67e-11	1.98e-12	3.67e-11	1.40e-13
2048	1.85e-13	3.91e-11	6.33e-11	1.35e-10	6.33e-11	1.35e-10	1.08e-13

Table 4 Maxima of the relative errors of the computed weights for n -point GQRs associated with the Chebyshev weight of the first kind $w(x) = (1 - x^2)^{-1/2}$. For each n , the smallest error is displayed in boldface

n	$\max_j \frac{ w_j - \omega_j }{ w_j }$						
	LMV	SymmAG	SymmAGWo	SymmAw	SymmMGWo	SymmMw	gauss
32	4.12e-14	1.73e-14	2.37e-14	4.55e-14	2.40e-14	4.55e-14	1.99e-14
64	5.13e-14	5.20e-14	8.34e-14	1.90e-13	8.35e-14	1.90e-13	4.21e-14
128	1.33e-12	6.26e-14	4.43e-13	2.24e-13	4.43e-13	2.24e-13	9.96e-14
256	3.40e-12	1.33e-13	3.15e-12	4.44e-13	3.15e-12	4.44e-13	6.66e-13
512	5.78e-12	1.45e-12	5.36e-11	4.93e-12	5.36e-11	4.93e-12	5.58e-13
1024	6.28e-12	1.54e-11	6.98e-11	4.94e-11	6.98e-11	4.94e-11	1.15e-11
2048	9.16e-11	4.88e-11	1.48e-09	1.75e-10	1.48e-09	1.75e-10	6.08e-11

We computed the Hermite weights, for different values of n , with the following three different methods:

- the function `gauss` [9], computed in double precision.
- the function `gauss` [9], computed in variable precision with 128 digits; these values can therefore be considered as exact values for the weights.
- the proposed method, called LMV.

In Fig. 2 (top) we plotted on a logarithmic scale the absolute values of the Hermite polynomials of degree $j = 0, 1, \dots, 127$, evaluated in the largest zeros of the Hermite polynomial of degree 128, computed by the function `gauss` in double and extended precision (128 digits) and by the proposed method in double precision. In Fig. 2 (bottom), we show the componentwise relative errors of each weight $e_{rel}(\omega_j^g) := \frac{|\omega_j^g - \omega_j^{ex}|}{|\omega_j^{ex}|}$, computed with `gauss` (in “+”), and $e_{rel}(\omega_j^{LMV}) :=$

Table 5 Maxima of the relative errors of the computed nodes for n -point GQRs associated with the Chebyshev weight of second kind $w(x) = (1 - x^2)^{1/2}$. For each n , the smallest error is displayed in boldface

n	$\max_j \frac{ \lambda_j - x_j }{ x_j }$						
	LMV	SymmAG	SymmAGWo	SymmAw	SymmMGWo	SymmMw	gauss
32	2.04e-15	6.12e-15	6.27e-15	5.10e-15	6.27e-15	5.10e-15	1.60e-15
64	5.64e-14	1.78e-14	8.62e-14	3.55e-14	8.62e-14	3.55e-14	2.88e-13
128	9.83e-15	1.45e-13	6.52e-14	7.18e-14	6.52e-14	7.18e-14	2.05e-14
256	1.70e-14	6.20e-13	9.67e-13	1.69e-12	9.67e-13	1.69e-12	4.11e-14
512	2.01e-14	2.55e-12	1.18e-12	1.16e-12	1.18e-12	1.16e-12	7.25e-14
1024	9.40e-14	1.02e-11	1.83e-12	1.75e-11	1.83e-12	1.75e-11	1.86e-13
2048	3.43e-14	4.15e-11	5.27e-11	4.35e-11	5.27e-11	4.35e-11	1.94e-13

Table 6 Maxima of the relative errors of the computed weights for n -point GQRs associated with the Chebyshev weight of second kind $w(x) = (1 - x^2)^{1/2}$. For each n , the smallest error is displayed in boldface

n	$\max_j \frac{ w_j - \omega_j }{ w_j }$						
	LMV	SymmAG	SymmAGWo	SymmAw	SymmMGWo	SymmMw	gauss
32	9.41e-15	1.53e-14	2.06e-14	7.11e-15	2.03e-14	7.11e-15	2.38e-14
64	1.80e-13	4.13e-14	1.31e-13	5.04e-14	1.31e-13	5.04e-14	2.43e-13
128	5.60e-13	8.57e-14	3.02e-13	4.91e-13	3.02e-13	4.91e-13	4.68e-13
256	2.52e-12	2.41e-13	3.81e-12	1.41e-12	3.81e-12	1.41e-12	6.59e-13
512	2.00e-12	2.44e-12	5.02e-12	7.16e-12	5.02e-12	7.16e-12	6.35e-12
1024	9.68e-12	3.48e-12	6.28e-11	2.01e-11	6.28e-11	2.01e-11	1.09e-11
2048	5.59e-11	1.78e-11	1.45e-10	6.59e-11	1.45e-10	6.59e-11	2.21e-11

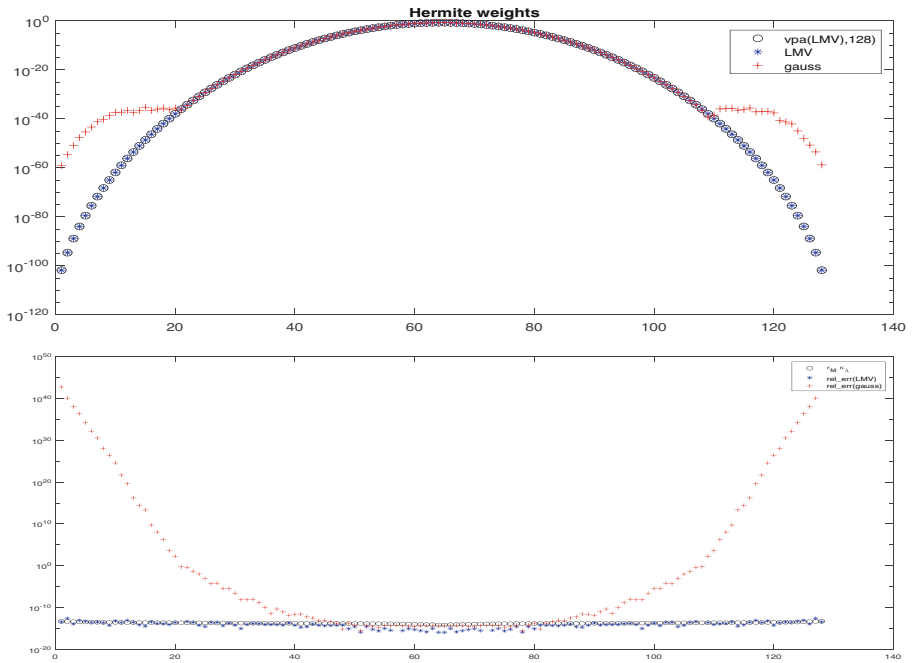


Fig. 2 Top: Plot of the Hermite weights for $n = 128$, computed with the function `gauss` in double precision (denoted by “+”) and in extended precision with 128 digits (denoted by “o”), and computed by the function `LMV` (denoted by “*”). Bottom: Plot of the componentwise relative errors of each weight $e_{rel}(\omega_j^g)$, computed with `gauss` (in “+”), and $e_{rel}(\omega_j^{LMV})$, computed with the new method `LMV` (in “*”). The corresponding relative error estimate $e_{est}(\omega_j^{LMV})$ for each weight computed with the method `LMV` is also given in “o”

Table 7 Approximation of the integral (21) obtained by an n -point GQR computed with the proposed method (second column) and with the function `gauss` (third column), for different values of n

$$\mathcal{I}(-0.8, 20) = 7.259087158081444e-02$$

n	LMV	<code>gauss</code>
16	7.086371480543081e-02	7.086371480542994e-02
32	7.258786172774918e-02	7.258786171871633e-02
64	7.259087115270522e-02	7.259907682924828e-02
128	7.259087158153937e-02	5.138844904420384e+22
256	7.259087158081003e-02	3.951943223796702e+75

$\frac{|\omega_j^{LMV} - \omega_j^{ex}|}{|\omega_j^{ex}|}$, computed with the new method LMV (in “*”). We also show the corresponding relative error estimate of each weight $e_{est}(\omega_j^{LMV}) := \frac{\epsilon_M \max_i |\lambda_i|}{\min_{i, i \neq j} |\lambda_i - \lambda_j|}$ (in “o”). We observe that the Hermite weights corresponding to large nodes are very tiny and `gauss`, which is based on the QR method, computes them with only normwise backward accuracy, but their elementwise relative forward error can be quite bad. On the other hand, LMV computes them with a relative forward error that is component-wise of the order of the machine precision, as predicted by our analysis in Section 6, and the estimated bounds of Theorem 5 are quite accurate.

Hence, integrals can not be approximated accurately with the nodes and weights provided by `gauss` if $f(x) \sim e^{cx^2}$, $1 - \varepsilon \leq c < 1$, with $\varepsilon > 0$, small enough.

Let us now consider the integral [13]

$$\mathcal{I}(a, b) = \int_{-\infty}^{\infty} e^{-(1+a)x^2 - \frac{b}{x^2}} dx = \sqrt{\frac{\pi}{1+a}} e^{-2\sqrt{(1+a)b}}, \quad a > -1, \quad b > 0, \quad (21)$$

that can be rewritten as

$$\mathcal{I}(a, b) = \int_{-\infty}^{\infty} e^{-x^2} e^{-ax^2 - \frac{b}{x^2}} dx = \int_{-\infty}^{\infty} \omega(x) e^{-ax^2 - \frac{b}{x^2}} dx.$$

If a is close to -1 , then the n -point GQR computed with `gauss` blows up as n increases. In Table 7 the approximation of the integral with GQR computed with the proposed method and with the function `gauss` are reported, for $a = -0.8$ and $b = 20$. The correct digits are highlighted in boldface.

8 Conclusions

The nodes and the weights of n -point Gaussian quadrature rules are computed from the eigenvalue decomposition of a tridiagonal matrix of order n by the Golub and Welsch algorithm. In case the weight function is symmetric, Meurant and Sommariva showed that the same information can be fetched from a tridiagonal matrix of order $\lfloor \frac{n}{2} \rfloor$, proposing different algorithms.

In this paper it is shown that, for symmetric weight functions, the positive nodes and the corresponding weights can be computed from a bidiagonal matrix with positive entries of size $\lfloor \frac{n}{2} \rfloor \times \lfloor \frac{n+1}{2} \rfloor$. Therefore the nodes can be computed with high relative accuracy by an algorithm proposed by Demmel and Kahan. Moreover, the stability of different methods for computing the weights is analyzed, proposing an algorithm for computing them with relative accuracy of the order of the machine precision. The numerical experiments confirm the effectiveness of the proposed approach.

Acknowledgements This work was supported partly by the Short Term Mobility program 2021 of Consiglio Nazionale delle Ricerche and partly by Gruppo Nazionale Calcolo Scientifico (GNCS) of Istituto Nazionale di Alta Matematica (INdAM).

The authors thank Gerard Meurant for the helpful suggestions and the anonymous referees for their constructive remarks.

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Croz, J.D.u., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D. LAPACK Users' Guide, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia (1999)
2. Bart, H., Gohberg, I., Kaashoek, M., Van Dooren, P.: Factorization of transfer functions. *SIAM J. Contr.* **18**(6), 675–696 (1980)
3. Bickley, W.G., Comrie, L.J., Sadler, D.H., Miller, J.C.P., Thompson, A.J.: British Association for the Advancement of Science Mathematical Tables: Volume 10, Bessel Functions, Part 2. Functions of Positive Integer Order. Cambridge University Press (1952)
4. Bowdler, H., Martin, R.S., Reinsch, C., Wilkinson, J.H.: The QR and QL algorithms for symmetric matrices. *Numer. Math.* **11**, 293–306 (1968)
5. Demmel, J., Kahan, W.: Accurate singular values of bidiagonal matrices. *SIAM J. Sci. Stat. Comput.* **11**(5), 873–912 (1990)
6. Fernando, K.V., Parlett, B.N.: Accurate singular values and differential qd algorithms. *Numer. Math.* **67**(2), 191–230 (1994)
7. Gautschi, W.: *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, Oxford (2004)
8. Gautschi, W.: Computational aspects of three-term recurrence relations. *SIAM Rev.* **9**, 24–82 (1967)
9. Gautschi, W.: Orthogonal polynomials (in Matlab). *J. Comput. Appl. Math.* **178**, 215–234 (2005)

10. Glaser, A., Liu, X., Rokhlin, V.: A fast algorithm for the calculation of the roots of special functions. *SIAM J. Sci. Comput.* **29**, 1420–1438 (2007)
11. Golub, G.H., Van Loan, C.F. *Matrix Computations*, 4th edn. Johns Hopkins University Press, Baltimore (2013)
12. Golub, G.H., Welsch, J.H.: Calculation of Gauss quadrature rules. *Math. Comput.* **23**(106), 221–230 (1969)
13. Gradshteyn, I.S., Ryzhik, I.M.: *Table of Integrals, Series, and Products*, 7th edn. Academic Press, Boston (2007)
14. Hale, N., Townsend, A.: Fast and accurate computation of Gauss-Legendre and Gauss-Jacobi nodes and weights. *SIAM J. Sci. Comput.* **35**, A652–A674 (2013)
15. Hale, N., Trefethen, L.N.: *Chebfun and numerical quadrature*. *Sci. China Math.* **55**(9), 1749–1760 (2012)
16. Higham, N.J. *Accuracy and Stability of Numerical Algorithms*, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2002)
17. Laudadio, T., Mastronardi, N., Van Dooren, P.: Computing the eigenvectors of nonsymmetric tridiagonal matrices. *Comput. Math. Math. Phys.* **61**, 733–749 (2021)
18. Mastronardi, N., Taeter, H., Van Dooren, P.: On Computing Eigenvectors of Symmetric Tridiagonal Matrices. In: Bini, D., Di Benedetto, F., Tyrtyshnikov, E., Van Barel, M. (eds.), vol. 30, pp. 181–195. Springer INdAM Series, Cham (2019)
19. Meurant, G., Sommariva, A.: Fast variants of the Golub and Welsch algorithm for symmetric weight functions in Matlab. *Numer. Algor.* **67**(3), 491–506 (2014)
20. Olver, F.W.J.: Error analysis of Miller’s recurrence equations. *Math. Comput.* **18**, 65–74 (1964)
21. Parlett, B.N., Le, J.: Forward instability of tridiagonal QR. *SIAM J. Matrix Anal. Appl.* **14**, 279–316 (1993)
22. Parlett, B.N.: *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs (1980)
23. Szegő, G. *Orthogonal Polynomials*, 4th edn. American Mathematical Society, Providence (1975)
24. Tait, R.: Error analysis of recurrence equations. *Math. Comput.* **21**, 629–638 (1967)
25. Townsend, A., Trogdon, T., Olver, S.: Fast computation of Gauss quadrature nodes and weights on the whole real line. *IMA J. Numer. Anal.* **36**, 337–358 (2016)
26. Wilf, H.S.: *Mathematics for the Physical Sciences*. Wiley, New York (1962)
27. Wilkinson, J.H.: *The Algebraic Eigenvalue Problem*. England, Oxford (1965)

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.