Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

ACCURATE COMPUTATIONS WITH TOTALLY POSITIVE MATRICES APPLIED TO THE COMPUTATION OF GAUSSIAN QUADRATURE FORMULAE[*]

ANA MARCO[†], JOSÉ-JAVIER MARTÍNEZ[†], AND RAQUEL VIAÑA[†]

Abstract. For some families of classical orthogonal polynomials defined on appropriate intervals, it is shown that the corresponding Jacobi matrices are totally positive and their bidiagonal factorizations can be accurately computed. By exploiting these facts, an algorithm to compute with high relative accuracy the eigenvalues of those Jacobi matrices, and consequently the nodes of Gaussian quadrature formulae for those families of orthogonal polynomials, is presented. An algorithm is also presented for the computation of the eigenvectors of these Jacobi matrices, and hence the weights of Gaussian quadrature formulae. Although in this case high relative accuracy is not theoretically guaranteed, the numerical experiments with our algorithm provide very accurate results.

Key words. Totally positive matrices, High relative accuracy, Gaussian quadrature, Orthogonal polynomials, Jacobi matrix, Eigenvalue computation, Eigenvector computation.

AMS subject classifications. 65F15, 5A23, 15B48, 65D30, 3C45.

1. Introduction.

The aim of this work is to show the role of the bidiagonal factorization of totally positive matrices (see [4] and [38] for general results on total positivity) in the construction of Gaussian quadrature formulae for some families of classical orthogonal polynomials, for which the corresponding symmetric Jacobi matrix is totally positive.

The design of accurate algorithms to work with structured totally positive matrices is an important topic of research in numerical linear algebra since, although problems in this field can be solved for such matrices by means of standard algorithms, by taking advantage of total positivity much more accurate results can be obtained (see, for example, [3, 26, 29]).

It is well known that the computation of the nodes and weights of a Gaussian quadrature formula is reduced to the calculation of the roots of an orthogonal polynomial, and that these roots can be obtained by computing the eigenvalues of its associated Jacobi matrix. This second point, which is a consequence of the three-term recurrence relation which orthogonal polynomials satisfy, is the basis of the widely known Golub–Welsch approach to calculate Gaussian quadrature formulae introduced in [16]. The connections between Gauss quadrature rules and the algebraic eigenvalue problem were already present in the book of Wilf [43], but the contribution of [16] is related to the numerical aspects, showing how to modify the QR algorithm for computing eigenvalues and eigenvectors in such a way that only the required components of the eigenvectors are computed. The importance and usefulness of the Golub–Welsch approach has been widely reported in the literature (see, for instance, [7, 41, 42]).

---

[*]Received by the editors on June 14, 2022. Accepted for publication on November 19, 2022. Handling Editor: Froilán Dopico. Corresponding Author: Ana Marco

[†]Departamento de Física y Matemáticas, Universidad de Alcalá, Campus Universitario, 28871-Alcalá de Henares (Madrid), Spain (ana.marco@uah.es, jjavier.martinez@uah.es, raquel.viana@uah.es). This research has been partially supported by Spanish Research Grant PGC2018-096321-B-I00 from the Spanish Ministerio de Ciencia, Innovación y Universidades. The authors are members of the Research Group ASYNACS (Ref.CT-CE2019/683) of Universidad de Alcalá.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

Although the approach we present in this work to compute Gaussian quadrature formulae is based on linear algebra, it is not the Golub–Welsch method nor any of the variants considered in [34], but an approach related to the representation of the positive definite Jacobi matrix J as $J = LDL^T$, which leads to the Cholesky factorization $J = R^T R$. The interest of this representation has been highlighted in different contexts in [36] and more recently in [35]. In connection with Gaussian quadrature, this approach for positive definite Jacobi matrices was suggested by Laurie [23, 24], who expressed this computations as a *coupled two-term recursion*.

Jacobi matrices are tridiagonal matrices, and if they are positive definite this implies a positive diagonal. Of course, there exist positive definite symmetric tridiagonal matrices with negative off-diagonal entries, but in the case of Jacobi matrices (as it will be seen in Section 2) the off-diagonal entries are $\sqrt{b_i}$ and so they are also positive. In this case, as recalled in [28] (p. 110), the Jacobi matrix is totally nonnegative (totally positive in the classical terminology of Pinkus [38]), and so we can use the algorithms of Plamen Koev [20] for totally nonnegative matrices. In addition, as we will see, we are in the positive case considered in [37], which should imply a better behavior of the corresponding algorithms.

In [28] the special case of the Gaussian quadrature formulae related to the Marchenko–Pastur measure was considered, but only the nodes were computed, since the accurate computation of the eigenvectors of the corresponding Jacobi matrices could not be achieved with MATLAB, which has led us to look for a new way to carry out this computation.

This problem has been clearly recalled by Gautschi at the end of the preface of [11]: *There is still a technical issue that should be mentioned. When some of the quadrature weights are very small (say, of the order* 10^{-50} *or even* 10^{-100}*), our* MATLAB *routine* `gauss.m` *or its variable-precision counterpart* `sgauss.m` *(both included in the dataset) may produce zero for these weights, apparently caused by the in-house* MATLAB *routine* `eig.m` *returning zero for very small eigenvector components.*

Recently, different interesting approaches to the construction of Gaussian quadrature formulae have appeared [13, 14, 17, 42], but those methods are not related to eigenproblems with Jacobi matrices. For instance, [14] is based on purely iterative methods, [13] on purely asymptotic methods, while [17, 42] employ iterative methods based on asymptotics.

In the context of the methods of [42], the authors indicate that *it is reasonable to use the Golub–Welsch algorithm to furnish initial guesses for the Newton's method*. The authors of [14] admit the Golub–Welsch approach is interesting for computing quadrature rules of low degree. So we think the numerical linear algebra approach is still of interest.

In fact, the very recent paper [21] considers the numerical linear algebra approach to compute Gaussian quadrature rules. In that paper, which mainly addresses the special case of symmetric weight functions, the authors indicate that the use of LAPACK subroutine `DLASQ1` (which was called by the algorithm TNEigenValues used in [28]) provides high relative accuracy in the computation of the nodes. As for the computation of the weights, the authors of [21] use an approach which is partly based on the eigenvector computation presented in [32]. Our approach to the computation of the weights, which is presented in detail in Section 4, is based on the computation of the right singular vectors of a bidiagonal matrix by means of the LAPACK routine `DBDSQR`.

The rest of the paper is organized as follows. The relationship between the Gaussian quadrature formulae, the orthogonal polynomials, and the Jacobi matrices is exposed briefly in Section 2. In Section 3, the

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

bidiagonal decomposition of several totally positive Jacobi matrices corresponding to shifted Chebyshev and Laguerre orthogonal polynomials are calculated. An algorithm that, given the bidiagonal decomposition of a Jacobi matrix J presented in Section 3, computes the eigenvalues and eigenvectors of J is developed in Section 4. Several numerical experiments showing the good performance of our approach are included in Section 5. Finally, Section 6 is devoted to the conclusions and some final comments.

2. Gaussian quadrature formulae, orthogonal polynomials, and Jacobi matrices.

Let $dw(t)$ be a nonnegative measure on the interval (a,b) such that the *moments*:

$$\mu_k = \int_a^b x^k dw(x), \quad k = 0,1,2,\dots$$

exist and are finite.

A classical problem in numerical analysis is to approximate the value of the integral:

$$\int_a^b f(x)dw(x)$$

by means of a *quadrature formula*:

$$\int_a^b f(x)dw(x) \approx w_1 f(x_1) + \cdots + w_n f(x_n),$$

where the *nodes* $\{x_i\}_{i=1}^n$ and the *weights* $\{w_i\}_{i=1}^n$ must be adequately selected.

A quadrature formula has *degree of exactness* d when it is exact for all polynomials of degree less than or equal to d. Using orthogonal polynomial theory, it is proved that the maximum degree of exactness of an n-point quadrature formula is $2n-1$ (see, for example, [39]). The *Gaussian quadrature formulae* are the quadrature formulae that yield this optimal degree of exactness.

The Gaussian quadrature formulae theory is based on the orthogonal polynomial theory. A good introduction to the orthogonal polynomial theory can be found in [2, 40] (see also Chapter 1 of [8]), but for the sake of completeness the basic points of this theory that we will need in this work are summarized below.

A sequence of polynomials $\{p_j\}_{j=0}^{\infty}$, where $p_n(x)$ has degree n, is known as an *orthogonal polynomial* sequence with respect to the scalar product:

$$(f,g) = \int_a^b f(x)g(x)dw(x)$$

associated to the measure $dw(x)$ if

$$(p_j, p_k) = 0, \quad j \neq k.$$

As it is well known, given an orthogonal polynomial sequence associated with one measure, each $p_n(x)$ is uniquely determined up to multiplication by a nonzero constant. An usual way to determine an unique orthogonal polynomial sequence associated with a measure is to enforce all the polynomials to be monic. We will proceed in this way and, from now on, all the orthogonal polynomials considered in this work will be monic.

The next three theorems (see [30, 40]) will help us in our aim of computing Gaussian quadrature formulae.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

Theorem 2.1. *Let $p_n(x)$ be the monic orthogonal polynomial of degree n. Then, the n roots of $p_n(x)$ are real, simple, and they belong to the interval (a, b).*

Theorem 2.2. *Let*

$$\int_a^b f(x)dw(x) \approx w_1 f(x_1) + \cdots + w_n f(x_n),$$

be a quadrature formula with degree of exactness $2n-1$. Then, the points $\{x_i\}_{i=1}^n$ are the zeros of the orthogonal polynomial $p_n(x)$ associated with the measure $dw(x)$ on the interval (a, b).

Theorem 2.3. *The weights $\{w_i\}_{i=1}^n$ of a Gaussian quadrature formula are all positive.*

Another fundamental point in the orthogonal polynomial theory is the following *three-term recurrence relation* that polynomials in sequence $\{p_j\}_{j=0}^\infty$ satisfy

$$\begin{aligned} &p_0(x) = 1, \quad p_1(x) = x - a_0, \\ &p_{k+1}(x) = (x - a_k)p_k(x) - b_k p_{k-1}(x), \quad k = 1, 2, \dots, \end{aligned} \tag{2.1}$$

where the coefficients of the recurrence relation are

$$a_k = \frac{(p_k, xp_k)}{(p_k, p_k)}, \quad k = 0, 1, \dots; \qquad b_k = \frac{(p_k, p_k)}{(p_{k-1}, p_{k-1})}, \quad k = 1, 2, \dots. \tag{2.2}$$

It is also usually defined as:

$$b_0 = \int_a^b dw(x),$$

i.e., the measure of the interval (a, b).

At this point, and by means of Theorem 2.2, we have reduced the problem of computing the nodes $\{x_i\}_{i=1}^n$ of a Gaussian quadrature formula to the problem of computing the roots of an n-degree orthogonal polynomial. Furthermore, taking into account Theorem 2.1, we also know that these roots are real, simple and in the open interval (a, b).

The following theorem of Golub and Welsch [16], which connects the three topics in the title of this section (Gaussian quadrature formulae, orthogonal polynomials, and Jacobi matrices), will give us an effective procedure for computing these roots, and also the weights $\{w_i\}_{i=1}^n$ of a Gaussian quadrature formula.

Theorem 2.4. *Let*

$$J = \begin{bmatrix} a_0 & \sqrt{b_1} & 0 & 0 & 0 \\ \sqrt{b_1} & a_1 & \sqrt{b_2} & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \sqrt{b_{n-2}} & a_{n-2} & \sqrt{b_{n-1}} \\ 0 & 0 & \dots & \sqrt{b_{n-1}} & a_{n-1} \end{bmatrix}$$

be the Jacobi matrix of order n built from the three-term recurrence relation (2.1). The nodes $\{x_i\}_{i=1}^n$ of the corresponding Gaussian quadrature formulae are the eigenvalues of J, and the weights $\{w_i\}_{i=1}^n$ are

$$w_i = b_0 v_{i1}^2, \quad i = 1, 2, \dots, n,$$

where v_{i1} is the first component of the normalized eigenvector v_i of J corresponding to the eigenvalue x_i, and

$$b_0 = \int_a^b dw(x).$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

This theorem will be key in our approach since it let us to reduce our initial problem, the computation of the nodes $\{x_i\}_{i=1}^n$ and weights $\{w_i\}_{i=1}^n$ of a Gaussian quadrature formula, to the computation of the eigenvalues and eigenvectors of a matrix whose nonzero entries are the coefficients a_i of the three-term recurrence relation (2.1) and the square roots of the coefficients b_i also in (2.1). Looking at (2.2), it is easily seen that $b_i > 0$ $(i = 1, 2, \ldots)$, and therefore $\sqrt{b_i}$ is always defined.

This expression of the weights related to the eigenvectors of J is a consequence of the confluent form of the Christoffel–Darboux identity (see Chapter 2 of [43], [16], and [30]).

3. Bidiagonal factorization of Jacobi matrices.

In the area of numerical linear algebra, it is known that if we have an accurate bidiagonal decomposition of a totally positive matrix many computations, such as, for example, eigenvalue and singular value computation, can be performed with high relative accuracy [18, 19].

From now on, given a totally positive matrix A, we will denote by $\mathcal{BD}(A)$ the matrix containing all the nontrivial entries of the bidiagonal decomposition of A [19]. This bidiagonal decomposition is related to the complete Neville elimination of the matrix A, which (when no row exchanges are needed, as it will happen in our case) consists of computing the Neville elimination of A and also of A^T. A detailed explanation of this fact can be found in [19, 27].

The aim of this section is the accurate computation of the bidiagonal decomposition of the Jacobi matrices corresponding to some classical orthogonal polynomials, specifically, the shifted Chebyshev on $[0, 1]$ (see Section 1.3 in [31]) and Laguerre polynomials. This accurate computation will allow us to calculate in Section 4 the eigenvalues and eigenvectors of these Jacobi matrices accurately, and therefore, to obtain the nodes and weights of the Gaussian quadrature formulae associated with these orthogonal polynomials accurately.

Taking into account that the Jacobi matrices J are not only tridiagonal, but also symmetric (see Theorem 2.4), and that the Neville elimination for triangularizing a tridiagonal symmetric matrix is the same as Gaussian elimination (without pivoting), the bidiagonal decomposition of J is $J = LDL^T$, where D is a diagonal matrix and L is a lower bidiagonal matrix with diagonal entries equal to 1. Consequently, the matrix $\mathcal{BD}(J)$ is also tridiagonal and symmetric and has as diagonal entries the diagonal entries of D, and as sub-diagonal and super-diagonal entries the sub-diagonal entries of L.

3.1. Bidiagonal decomposition of Jacobi matrices associated with shifted Chebyshev polynomials on $[0, 1]$.

We start by taking the measure function $dw(x) = (1-x)^{-1/2}x^{-1/2}dx$ on the interval $[0, 1]$. The associated monic orthogonal polynomials are the monic *shifted Chebyshev polynomials of the first kind* on the interval $[0, 1]$, and their corresponding Jacobi matrix of order n (see the algorithm `r_jacobi01` in the package OPQ of W. Gautschi [12]) is

$$J = tridiag\left[\begin{array}{ccccccccccc} & \frac{\sqrt{2}}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} & \\ \frac{1}{2} & & \frac{1}{2} & & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} \\ & \frac{\sqrt{2}}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} & \end{array}\right]. \tag{3.3}$$

The theorem below gives the bidiagonal decomposition of J.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

THEOREM 3.1. *The Jacobi matrix J in equation (3.3) is totally positive and its bidiagonal decomposition is given by the tridiagonal matrix:*

$$\mathcal{BD}(J) = tridiag\left[\begin{array}{ccccccccccc} & \frac{\sqrt{2}}{2} & & 1 & & 1 & & \cdots & & 1 & \\ \frac{1}{2} & & \frac{1}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} \\ & \frac{\sqrt{2}}{2} & & 1 & & 1 & & \cdots & & 1 & \end{array}\right]. \tag{3.4}$$

Proof. By carring out the Neville elimination on J (which, as we have indicated before, in the symmetric tridiagonal case is the same as Gaussian elimination) no row and column exchanges are needed, and it is seen that the nontrivial entries of D in $J = LDL^T$ are $d_{1,1} = \frac{1}{2}$ and $d_{i,i} = \frac{1}{4}$ $(i = 2, \dots, n)$, and that the sub-diagonal entries in L are $l_{2,1} = \frac{\sqrt{2}}{2}$ and $l_{i,i-1} = 1$ $(i = 3, \dots, n)$. Consequently, J is totally positive [6] and its bidiagonal decomposition is given by the tridiagonal matrix $\mathcal{BD}(J)$ in the statement of this theorem.

Let us observe that the total positivity of J can also be deduced from the fact that J is a positive definite tridiagonal matrix (see Theorem 2.1 and Theorem 2.4) whose all off-diagonal entries are nonnegative [1, 38]. $\square$

Now let us consider the measure function $dw(x) = (1-x)^{1/2}x^{1/2}dx$ on the interval $[0, 1]$ and its associated monic orthogonal polynomials, that is, the monic *shifted Chebyshev polynomials of the second kind* on the interval $[0, 1]$. Their corresponding Jacobi matrix of order n is (see `r_jacobi01` in the package OPQ of W. Gautschi [12])

$$J = tridiag\left[\begin{array}{ccccccccccc} & \frac{1}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} & \\ \frac{1}{2} & & \frac{1}{2} & & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} \\ & \frac{1}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} & \end{array}\right], \tag{3.5}$$

and its bidiagonal decomposition is given by the following theorem. Its proof and the proofs of the other theorems in this section are analogous to the one of Theorem 3.1.

THEOREM 3.2. *The Jacobi matrix in (3.5) is totally positive and its bidiagonal decomposition is contained in the tridiagonal matrix:*

$$\mathcal{BD}(J) = tridiag\left[\begin{array}{ccccccccccc} & \frac{1}{2} & & \frac{2}{3} & & \frac{3}{4} & & \cdots & & \frac{n-1}{n} & \\ \frac{1}{2} & & \frac{3}{8} & & \frac{4}{12} & & \frac{5}{16} & & \cdots & & \frac{n+1}{4n} \\ & \frac{1}{2} & & \frac{2}{3} & & \frac{3}{4} & & \cdots & & \frac{n-1}{n} & \end{array}\right]. \tag{3.6}$$

Then we take the monic *shifted Chebyshev polynomials of the third kind* on the interval $[0, 1]$, i.e., the monic orthogonal polynomials associated with the measure function $dw(x) = (1-x)^{-1/2}x^{1/2}dx$ on the interval $[0, 1]$. As it is known (see `r_jacobi01` in the package OPQ of W. Gautschi [12]), the Jacobi matrix associated to these polynomials is

$$J = tridiag\left[\begin{array}{ccccccccccc} & \frac{1}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} & \\ \frac{3}{4} & & \frac{1}{2} & & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} \\ & \frac{1}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} & \end{array}\right]. \tag{3.7}$$

The next theorem provides the explicit expressions of the nontrivial entries in the bidiagonal decomposition of J.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

THEOREM 3.3. *The Jacobi matrix presented in* (3.7) *is totally positive and its bidiagonal decomposition is given by the tridiagonal matrix:*

$$\mathcal{BD}(J) = tridiag\left[\begin{array}{cccccccc} & \frac{1}{3} & & \frac{3}{5} & & \frac{5}{7} & & \cdots & & \frac{2n-3}{2n-1} & \\ \frac{3}{4} & & \frac{5}{12} & & \frac{7}{20} & & \frac{9}{28} & & \cdots & & \frac{2n+1}{4(2n-1)} \\ & \frac{1}{3} & & \frac{3}{5} & & \frac{5}{7} & & \cdots & & \frac{2n-3}{2n-1} & \end{array}\right]. \tag{3.8}$$

Finally, we consider the monic *shifted Chebyshev polynomials of the fourth kind* on the interval $[0,1]$, that is to say, the monic orthogonal polynomials associated with the measure function $dw(x) = (1-x)^{1/2}x^{-1/2}dx$ on the interval $[0,1]$. Their associated Jacobi matrix is (see `r_jacobi01` in the package OPQ of W. Gautschi [12]):

$$J = tridiag\left[\begin{array}{ccccccccccc} & \frac{1}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} & \\ \frac{1}{4} & & \frac{1}{2} & & \frac{1}{2} & & \frac{1}{2} & & \cdots & & \frac{1}{2} \\ & \frac{1}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} & \end{array}\right], \tag{3.9}$$

and its bidiagonal decomposition is provided by the following theorem.

THEOREM 3.4. *The Jacobi matrix in equation* (3.9) *is totally positive and its bidiagonal decomposition is given by the tridiagonal matrix:*

$$\mathcal{BD}(J) = tridiag\left[\begin{array}{ccccccccccc} & 1 & & 1 & & 1 & & \cdots & & 1 & \\ \frac{1}{4} & & \frac{1}{4} & & \frac{1}{4} & & \frac{1}{4} & & \cdots & & \frac{1}{4} \\ & 1 & & 1 & & 1 & & \cdots & & 1 & \end{array}\right]. \tag{3.10}$$

3.2. Bidiagonal decomposition of Jacobi matrices associated with Laguerre polynomials. Let $dw(x)$ be the measure function on the interval $[0,+\infty)$, where $w(x) = e^{-x}$. As it is well known (see, for instance, [8]) the corresponding monic orthogonal polynomials are the monic *Laguerre polynomials* and their associated Jacobi matrix of order n is

$$J = tridiag\left[\begin{array}{ccccccccccc} & 1 & & 2 & & 3 & & \cdots & & n-1 & \\ 1 & & 3 & & 5 & & 7 & & \cdots & & 2n-1 \\ & 1 & & 2 & & 3 & & \cdots & & n-1 & \end{array}\right]. \tag{3.11}$$

The following theorem provides the exact bidiagonal decomposition of J. Its proof and the one of the other theorem in this section are similar to the one of Theorem 3.1.

THEOREM 3.5. *The Jacobi matrix J in* (3.11) *is totally positive and its bidiagonal decomposition is given by the tridiagonal matrix:*

$$\mathcal{BD}(J) = tridiag\left[\begin{array}{ccccccccccc} & 1 & & 1 & & 1 & & \cdots & & 1 & \\ 1 & & 2 & & 3 & & 4 & & \cdots & & n \\ & 1 & & 1 & & 1 & & \cdots & & 1 & \end{array}\right]. \tag{3.12}$$

Finally, we consider the monic *generalized Laguerre polynomials*, that is, the orthogonal polynomials associated with the measure function $dw(x) = x^{\alpha}e^{-x}dx$, where $\alpha > -1$, on the interval $[0,\infty)$. Their corresponding Jacobi matrix J of order n is [8]

$$J = tridiag\left[\begin{array}{ccccccccc} & \sqrt{1+\alpha} & & \sqrt{2(2+\alpha)} & & \cdots & & \sqrt{(n-1)(n-1+\alpha)} & \\ 1+\alpha & & 3+\alpha & & 5+\alpha & & \cdots & & 2n-1+\alpha \\ & \sqrt{1+\alpha} & & \sqrt{2(2+\alpha)} & & \cdots & & \sqrt{(n-1)(n-1+\alpha)} & \end{array}\right]. \tag{3.13}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

The next theorem gives the explicit expressions for the nontrivial entries in the bidiagonal factorization of J.

Theorem 3.6. *The Jacobi matrix J in equation (3.13) is totally positive and its bidiagonal decomposition is contained in the tridiagonal matrix:*

$$\mathcal{BD}(J) = tridiag\left[\begin{array}{ccccccccc} & \sqrt{\frac{1}{1+\alpha}} & & \sqrt{\frac{2}{2+\alpha}} & & \cdots & & \sqrt{\frac{n-1}{n-1+\alpha}} & \\ 1+\alpha & & 2+\alpha & & 3+\alpha & & \cdots & & n+\alpha \\ & \sqrt{\frac{1}{1+\alpha}} & & \sqrt{\frac{2}{2+\alpha}} & & \cdots & & \sqrt{\frac{n-1}{n-1+\alpha}} & \end{array}\right]. \tag{3.14}$$

Looking at the expressions of the $\mathcal{BD}(J)$ matrices obtained in this section (see (3.4), (3.6), (3.8), (3.10), (3.12), and (3.14)), we notice that not only we can compute these matrices with high relative accuracy, but we can also compute them exactly. This fact will be of great importance in our approach, since the computation of $\mathcal{BD}(J)$ is the first stage in our method to compute the eigenvalues and eigenvectors of the Jacobi matrices introduced in this section, and consequently, to the accurate computation of the nodes and weights of the quadrature formulae for the corresponding measures (see Section 4). The importance of accurately computing the entries of the Jacobi matrices to obtain accurate quadrature formulae has also been recently pointed out in [25].

4. Accurate computation of the eigenvalues and eigenvectors of Jacobi matrices.

In this section, we present an algorithm that we have called `MMVTNEigenVectors`, to compute the eigenvalues and eigenvectors of a totally positive Jacobi matrix J starting from its bidiagonal decomposition $\mathcal{BD}(J)$ (see Algorithm 1).

Given a Jacobi matrix J which is symmetric and positive definite, let us see how a singular value computation is an alternative here. Let $J = R^T R$ be the Cholesky factorization of J, and let $R = U\Sigma V^T$ be the singular value decomposition of R (where U and V are orthogonal matrices, and Σ is a diagonal matrix with the singular values of R as diagonal entries). Then, we have

$$J = R^T R = V\Sigma U^T U\Sigma V^T = V\Sigma^2 V^T.$$

Since V is an orthogonal matrix, we have

$$JV = V\Sigma^2,$$

which means the columns of V (i.e., the right singular vectors of R) are the eigenvectors of J (and, of course, the eigenvalues of J are the squares of the singular values of R). In fact, Laurie in Section 5 of [24] gives the alternative of using the function `svd` of Matlab to compute the eigenvalues of J, and the same use of `svd` is included in the algorithm `TNEigenValues` of P. Koev for computing the eigenvalues of a totally positive matrix [18, 20].

Algorithm 1 to compute the eigenvalues and eigenvectors of J is given in Matlab code. The computation of the eigenvalues of J (lines 3-5 in Algorithm 1) is carried out by means of an adaptation of the P. Koev's general algorithm `TNEigenValues`, taking into account that in our case J is a tridiagonal, symmetric, and totally nonnegative matrix. The input argument of `TNEigenValues` is $B = \mathcal{BD}(J)$, the matrix representing the bidiagonal decomposition of J. As recalled in [28], for the tridiagonal case $B = \mathcal{BD}(J)$ contains the entries of the factorization $J = LDL^T$, and so the Cholesky factor of J is computed as $R = \sqrt{D}L^T$. Thus, taking advantage of the specific structure of J, we have been able to eliminate some of the computations

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

that `TNEigenValues` does, and our algorithm only computes, starting from $\mathcal{BD}(J)$, the Cholesky factor R of J ($J = R^T R$), and its singular values by using the LAPACK routine `DLASQ1` [18, 22], based on [5].

We must observe that the algorithm `TNBD` of P. Koev, which gives $B = \mathcal{BD}(J)$, does not guarantee high relative accuracy, and the same happens with standard algorithms for computing the Cholesky decomposition of J. In our algorithm, we use as input the exact expressions for B, providing an approach which computes with high relative accuracy all the eigenvalues of the Jacobi matrices J considered in Section 3, and in consequence, the nodes $\{x_i\}_{i=1}^n$ of the Gaussian quadrature formulae corresponding to the measures in that section.

As for the computational cost of the eigenvalue computation process, it is leaded by the cost of the LAPACK routine `DLASQ1`, and consequently it is of $O(n^2)$ arithmetic operations [18], being n the order of the Jacobi matrix.

Algorithm 1 `MMVTNEigenVectors`

```
Require: $B$, the matrix $n \times n$ containing the bidiagonal decomposition $\mathcal{BD}(J)$ of $J$.
Ensure: The eigenvalues $a$ of $J$ ($a \in \mathbb{R}^n$) and the normalized eigenvectors of $J$ (the columns of $V \in \mathbb{R}^{n\times n}$).
1:  function $[a, V]$=MMVTNEigenVectors($B$)
2:  $n$=size(B);
3:  $d$=sqrt(diag(B));
4:  $e$=diag(B,1).*d(1:length(d)-1);
5:  $a$=mexdlasq1(d,e).^2;
6:  if nargout==2
7:      $[\sim, VT]$=mexdbdsqr(d,e);
8:      $V = VT'$;
9:      for i=1:n
10:         V(:,i)=V(:,i)/norm(V(:,i));
11:     end;
12: end
```

The computation of the eigenvectors of J (lines 6-12 in Algorithm 1) is accomplished by computing the right singular vectors of the Cholesky factor R of J by means of the LAPACK routine `DBDSQR` [22], based on [5]. As we have noted before, the eigenvectors of J are the right singular vectors of R.

The LAPACK routine `DBDSQR` computes the right singular vectors of a bidiagonal matrix by using the implicit zero-shift QR algorithm, and therefore its computational cost is of $O(n^3)$ arithmetic operations. So, the computational cost of the computation of the eigenvectors of J dominates the computational cost of `MMVTNEigenVectors` and in consequence, it is $O(n^3)$ arithmetic operations.

As for the accuracy of this stage of our algorithm, we have to say that although the LAPACK routine `DBDSQR` does not guarantee high relative accuracy in the right singular vector computation, it returns very accurate results as the experiments included in Section 5 show.

5. Numerical experiments.

This section includes two numerical experiments that illustrate the good behavior of the algorithm introduced in the previous section when applied to the calculation of the nodes and weights of Gaussian quadrature formulae. In the first one, tests in which measures whose associated orthogonal polynomials are shifted Chebyshev polynomials on $[0, 1]$ are considered, while in the second,

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

cases in which measures whose corresponding orthogonal polynomials are Laguerre and generalized Laguerre orthogonal polynomials are analyzed.

Example 5.1. *Let us consider the Gaussian quadrature formulae:*

$$\int_0^1 f(x)dw(x) \approx w_1 f(x_1) + \cdots + w_n f(x_n),$$

where different values of n and different measures $dw(x)$ are taken. Specifically, we consider $n = 64, 128, 256$, and the measures on $[0,1]$ introduced in Section 3.1, namely $dw(x) = (1-x)^{-1/2}x^{-1/2}dx$, $dw(x) = (1-x)^{1/2}x^{1/2}dx$, $dw(x) = (1-x)^{-1/2}x^{1/2}dx$ and $dw(x) = (1-x)^{1/2}x^{-1/2}dx$. Their associated monic orthogonal polynomials are the shifted Chebyshev polynomials of the first kind, the shifted Chebyshev polynomials of the second kind, the shifted Chebyshev polynomials of the third kind, and the shifted Chebyshev polynomials of the fourth kind, respectively.

We compute the nodes $\{x_i\}_{i=1}^n$ and the weights $\{w_i\}_{i=1}^n$ in each case by using the algorithm `MMVTNEigenVectors` introduced in Section 4 to compute the eigenvalues and eigenvectors of the corresponding Jacobi matrix J, starting from the matrix $\mathcal{BD}(J)$ containing the nontrivial elements of the bidiagonal factorization of J. The expressions of the matrices J, as well as the expressions of the matrices $\mathcal{BD}(J)$, are the ones in Section 3.1. Taking into account Theorem 2.4, the nodes $\{x_i\}_{i=1}^n$ are the eigenvalues of J and the weights $w_i = b_0 v_{i1}^2$, where v_{i1} is the first component of the normalized eigenvector v_i of J corresponding to the eigenvalue x_i and $b_0 = \int_a^b dw(x)$.

The values of b_0 are in this case $b_0 = \pi$, $b_0 = \pi/8$, $b_0 = \pi/2$ and $b_0 = \pi/2$, respectively.

We compare the results obtained by our approach with the ones obtained when using:

- The standard command `eig` from Matlab to compute the eigenvalues and eigenvectors of a matrix.
- The command `gauss` in the package OPQ of W. Gautschi [12] (see also [9, 10]) which starting from the coefficients of the three-term recurrence relation for the monic orthogonal polynomials computes the nodes and the weights of the Gaussian quadrature formula. This command calls `eig` from Matlab.
- The LAPACK routine `DPTEQR` to compute the eigenvalues and eigenvectors of a symmetric positive definite tridiagonal matrix.

In order to compare the relative errors obtained when computing the nodes and the weights by these four methods, we use the eigenvalues and eigenvectors computed with *Mathematica* by means of the commands `Eigenvalues` and `Eigenvectors` with 100 digits of numerical precision.

The results corresponding to the node computation are given in Table 1. Taking into account that classical algorithms for calculating eigenvalues of ill-conditioned totally positive matrices only compute the largest eigenvalues with guaranteed relative accuracy, whereas the tiny eigenvalues may be computed with no relative accuracy at all [18], only the relative error obtained in the computation of the smallest node of each test is included in Table 1.

The results of the computation of the weights are presented in Table 2, where only the smallest weight is included.

Let us observe here that, as the results obtained when using `gauss` and `eig` are practically the same, which is logical because `gauss` calls `eig`, only the results obtained when using one of the two commands (`gauss`) are included in Table 1 and Table 2.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

The ill-conditioning of the matrices considered in this example is shown in Table 3, where the two-norm condition numbers of these matrices computed in *Mathematica* are included.

Table 1
Relative error of the smallest node in Example 5.1.

Chebyshev	n	Smallest node	MMV	gauss	DPTEQR
1st	64	$1.5e-04$	$1.8e-16$	$2.4e-14$	$5.8e-15$
	128	$3.8e-05$	$5.4e-16$	$6.9e-13$	$1.1e-14$
	256	$9.4e-06$	0	$4.0e-12$	$2.2e-14$
2nd	64	$5.8e-04$	$7.4e-16$	$7.0e-14$	$9.1e-15$
	128	$1.5e-04$	$5.5e-16$	$3.2e-13$	$8.6e-15$
	256	$3.7e-05$	$2.0e-15$	$1.3e-12$	$2.6e-14$
3rd	64	$5.9e-04$	$1.8e-16$	$3.8e-14$	$5.5e-16$
	128	$1.5e-04$	$1.8e-16$	$1.6e-13$	$9.1e-15$
	256	$3.8e-05$	$1.4e-15$	$6.4e-13$	$9.2e-14$
4th	64	$1.5e-04$	$1.8e-16$	$2.9e-13$	$1.8e-16$
	128	$3.7e-05$	$3.6e-16$	$1.2e-12$	$3.6e-16$
	256	$9.4e-06$	0	$5.1e-12$	0

Table 2
Relative error of the smallest weight in Example 5.1.

Chebyshev	n	Smallest weight	MMV	gauss	DPTEQR
1st	64	$4.9e-02$	$2.8e-15$	$2.0e-14$	$5.8e-15$
	128	$2.5e-02$	$5.5e-15$	$5.8e-14$	$1.2e-14$
	256	$1.2e-02$	$1.8e-14$	$1.4e-13$	$2.0e-14$
2nd	64	$2.8e-05$	$2.2e-15$	$2.4e-14$	$1.5e-14$
	128	$3.6e-06$	$9.2e-14$	$2.2e-14$	$1.5e-14$
	256	$4.6e-07$	$3.9e-14$	$7.3e-14$	$8.8e-14$
3rd	64	$2.9e-05$	$7.4e-15$	$5.9e-14$	$3.4e-15$
	128	$3.7e-06$	$8.2e-14$	$5.5e-14$	$2.2e-14$
	256	$4.6e-07$	$3.3e-13$	$5.0e-14$	$6.6e-14$
4th	64	$2.9e-05$	$5.7e-13$	$2.7e-14$	$9.1e-13$
	128	$3.7e-06$	$1.2e-11$	$4.8e-13$	$1.0e-11$
	256	$4.6e-07$	$5.3e-11$	$6.6e-13$	$2.6e-12$

Example 5.2. *In this case, the Gaussian quadrature formulae:*

$$\int_0^{+\infty} f(x)dw(x) \approx w_1 f(x_1) + \cdots + w_n f(x_n),$$

are considered. As in Example 5.1, the values $n = 64, 128, 256$ and different measures $dw(x)$ are taken. In particular, the measures on $[0, +\infty)$ $dw(x) = x^{\alpha}e^{-x}dx$ with $\alpha = 0$, the Laguerre case, and $\alpha = 0.9, -0.9, -0.99$, the generalized Laguerre case, are now conssidered. Their corresponding Jacobi matrices J are the ones displayed in Section 3.2, and the tridiagonal matrices $\mathcal{BD}(J)$ containing the nontrivial

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

Table 3
Condition number of the matrices in Example 5.1.

Chebyshev	n	κ_2
1st	64	$6.6e+03$
	128	$2.7e+04$
	256	$1.1e+05$
2nd	64	$1.7e+03$
	128	$6.7e+03$
	256	$2.7e+04$
3rd	64	$1.7e+03$
	128	$6.7e+03$
	256	$2.7e+04$
4th	64	$6.7e+03$
	128	$2.7e+04$
	256	$1.1e+05$

elements of the bidiagonal decomposition of the matrices J are the ones obtained in the same section. The values of b_0 are $b_0 = 1$ in the Laguerre case, and $\Gamma(1+\alpha)$ in the generalized Laguerre case.

We compute the nodes $\{x_i\}_{i=1}^n$ and the weights $\{w_i\}_{i=1}^n$ in each case by means of the same approaches used in Example 5.1, and also by the Golub–Welsch approach by using the commands `mmq_classsicorthopoly` and `mmq_gaussquadrule` in the MMQ Toolbox (Matlab software for the book [15]) whose implementation in Matlab can be taken from [33]. `mmq_classsicorthopoly` supplies the coefficients of the normalized recurrences for various classical orthogonal polynomials, among which are Laguerre and generalized Laguerre but not shifted Chebyshev on $[0,1]$, and the corresponding moment of order zero. `mmq_gaussquadrule` is the Golub–Welsch algorithm and computes the nodes and the weights of the Gaussian quadrature formula from the output of `mmq_classsicorthopoly`.

As recalled in [34], this genuine Golub–Welsch approach means that only the first components of the eigenvectors are computed.

The results obtained in the node computation are shown in Table 4, and the results got in the computation of the weights are given in Table 5. Table 6 is devoted to the 2-norm condition numbers computed in *Mathematica* of the matrices considered in this example and illustrates the ill-conditioning of the matrices.

As in Example 5.1, the results obtained when using `eig` and `gauss` are equivalent and only the results of `gauss` are included in Table 4 and Table 5.

Looking at the results of Example 5.1 and Example 5.2 shown in Table 1 and Table 4, respectively, we note that our approach computes the nodes in both examples very accurately, in fact it is the algorithm that gives more accurate results. As for the computation of the weights, from the results in Table 2 and Table 5 we observe that our algorithm computes them in an accurate way. Special mention deserves the Laguerre case (Table 5) where, although the weights are really small in size, the accuracy of our approach is maintained while `gauss` and `DPTEQR` give results not accurate at all.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

TABLE 4
Relative error of the smallest node in Example 5.2.

α	n	Smallest node	`MMV`	`GW`	`gauss`	`DPTEQR`
0	64	$2.2e-02$	$3.1e-16$	$1.3e-13$	$2.9e-14$	$1.9e-15$
	128	$1.1e-02$	$7.7e-16$	$5.4e-13$	$5.4e-13$	$3.1e-15$
	256	$5.6e-03$	$1.2e-15$	$2.5e-12$	$3.4e-12$	$5.4e-15$
0.9	64	$5.3e-02$	$9.2e-16$	$5.1e-14$	$1.3e-14$	$9.4e-15$
	128	$2.6e-02$	$3.9e-16$	$3.9e-14$	$3.3e-13$	$9.0e-14$
	256	$1.3e-02$	$3.7e-15$	$2.9e-12$	$5.9e-13$	$3.4e-13$
-0.9	64	$1.6e-03$	$1.5e-15$	$6.6e-14$	$5.5e-14$	$4.2e-15$
	128	$8.2e-04$	$7.9e-16$	$3.3e-13$	$2.3e-13$	$2.1e-13$
	256	$4.1e-04$	$5.6e-15$	$4.6e-12$	$2.0e-12$	$2.5e-13$
-0.99	64	$1.6e-04$	$2.2e-15$	$7.1e-13$	$4.5e-14$	$2.4e-14$
	128	$7.9e-05$	$3.5e-16$	$1.7e-12$	$1.7e-13$	$1.0e-14$
	256	$3.9e-05$	$1.1e-14$	$2.1e-11$	$3.8e-12$	$5.7e-13$

TABLE 5
Relative error of the smallest weight in Example 5.2.

α	n	Smallest weight	`MMV`	`GW`	`gauss`	`DPTEQR`
0	64	$2.1e-101$	$1.2e-13$	$4.3e-14$	$8.6e+42$	$8.6e+67$
	128	$8.6e-210$	$1.5e-12$	$3.2e-13$	1	$2.4e+174$
	256	$1.1e-428$	$4.8e-13$	1	1	$1.7e+394$
0.9	64	$5.0e-100$	$3.5e-13$	$3.9e-14$	$6.1e+41$	$2.0e+65$
	128	$3.9e-208$	$9.6e-13$	$2.4e-13$	1	$9.1e+173$
	256	$9.6e-427$	$7.9e-14$	1	1	$5.6e+196$
-0.9	64	$8.8e-103$	$1.6e-13$	$1.7e-13$	$6.0e+44$	$1.8e+69$
	128	$1.9e-211$	$1.4e-13$	$4.4e-14$	1	$1.6e+174$
	256	$1.4e-430$	$1.3e-12$	1	1	$7.2e+194$
-0.99	64	$6.5e-103$	$2.1e-15$	7.8e-14	$1.2e+45$	$8.7e+69$
	128	$1.3e-211$	$1.4e-13$	$5.9e-15$	1	$6.8e+177$
	256	$8.8e-431$	$6.8e-13$	1	1	$3.0e+193$

REMARK 5.3. *Let us note that the values of the smallest weights for $n = 256$ in Table 5, and also the values computed by using our approach, are smaller than* `realmin` *(i.e., $2.23e-308$, the smallest normalized positive floating point number in double precision). This phenomenon is found for Gauss–Hermite quadrature weights in Section 3.6 of [42].*

To obtain them by means of our approach, we have computed the eigenvectors by using the algorithm `MMVTNEigenVectors` *in* MATLAB*, and then we have computed the weights in Mathematica by using the formula for the weights in Theorem 2.4.*

6. Conclusions and final comments.

In this work, an accurate algorithm to compute the nodes and weights of Gaussian quadrature formulae for the shifted Chebyshev on $[0, 1]$ case and the Laguerre

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

Table 6
Condition number of the matrices in Example 5.2.

α	n	κ_2
0	64	$1.0e+04$
	128	$4.3e+04$
	256	$1.8e+05$
0.9	64	$4.5e+03$
	128	$1.8e+04$
	256	$7.5e+04$
-0.9	64	$1.4e+05$
	128	$5.9e+05$
	256	$2.4e+06$
-0.99	64	$1.4e+06$
	128	$6.1e+06$
	256	$2.5e+07$

case is presented. The good results of our approach are a direct consequence of the explicit expressions we have obtained for the bidiagonal decomposition of the Jacobi matrices corresponding to these orthogonal polynomials. Such expressions allow us, on one hand, to prove the total positivity of the Jacobi matrices, and on the other hand, to compute accurately their bidiagonal decomposition. This stage is essential to obtain the eigenvalues and eigenvectors of the Jacobi matrices, and therefore, the nodes and the weights of the associated Gaussian quadrature formulas accurately.

The numerical experiments corroborate the accuracy of our algorithm. Referring to the nodes, the comparison with other available methods based on linear algebra indicates that our algorithm always provides the most accurate results. As for the weights, it can be observed that our algorithm gives always accurate results. In particular, it is worth noting that when the weights are close to zero, our method is still accurate while the other methods give really inaccurate results.

Acknowledgment. The authors are grateful to Dr. David Bindel for his advice on using the LAPACK routine `DBDSQR`. They also express their gratitude to the anonymous referees for their constructive comments and suggestions, which improved the quality of the paper.

REFERENCES

[1] A. Barreras and J. M. Peña. Characterizations of Jacobi sign regular matrices. *Linear Algebra Appl.*, 436: 381–388, 2012.
[2] T. S. Chihara. *An introduction to orthogonal polynomials.* Gordon and Breach, New York, 1978.
[3] J. Delgado, H. Orera, and J. M. Peña. Accurate algorithms for Bessel matrices. *J. Sci. Comput.*, 80:1264–1278, 2019.
[4] S. M. Fallat and C. R. Johnson. *Totally Nonnegative Matrices.* Princeton University Press, Princeton, NJ, 2011.
[5] K. V. Fernando and B. N. Parlett. Accurate singular values and differential quotient-difference algorithms. *Numer. Math.*, 67: 191–229, 1994.
[6] M. Gasca and J. M. Peña. On Factorizations of Totally Positive Matrices. In: Gasca, M. and Michelli, C. A. (Eds.), *Total Positivity and Its Applications (Jaca, 1994).* Kluwer Academic Publishers, Dordrecht, pp. 109–130, 1996.
[7] W. Gautschi. The interplay between classical analysis and (numerical) linear algebra – a tribute to Gene H. Golub. *Electr. Trans. Numer. Anal.* 13:119–147, 2002.
[8] W. Gautschi. *Orthogonal Polynomials. Computation and Approximation.* Oxford University Press, Oxford, 2004.
[9] W. Gautschi. Orthogonal Polynomials (in Matlab). *J. Comput. Appl. Math.*, 178:215–234, 2005.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 38, pp. 777-791, November 2022.

[10] W. Gautschi. *Orthogonal Polynomials in MATLAB: Exercises and Solutions.* SIAM, 2016.
[11] W. Gautschi. A software repository for Gaussian quadratures and Christoffel functions. *Software, Environments, and Tools, 32.* SIAM, Philadelphia, 2021.
[12] `https://www.cs.purdue.edu/archives/2002/wxg/codes/OPQ.html`
[13] A. Gil, J. Segura, and N. M. Temme. Asymptotic approximations to the nodes and weights of Gauss-Hermite and Gauss-Laguerre quadratures. *Stud. Appl.Math.*, 140(3):298–332, 2018.
[14] A. Gil, J. Segura, and N. M. Temme. Fast, reliable and unrestricted iterative computation of Gauss-Hermite and Gauss-Laguerre quadratures. *Numerische Mathematik* 143:649–682, 2019.
[15] G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications.* Princeton University Press, Princeton, New Jersey, 2010.
[16] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Math. Comp.*, 23:221–230, 1969.
[17] N. Hale and A. Townsend. Fast and accurate computation of Gauss-Legendre and Gauss-Jacobi quadrature nodes and weights. *SIAM J. Sci. Comput.*, 35(2):A652–A674, 2013.
[18] P. Koev. Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.*, 27:1–23, 2005.
[19] P. Koev. Accurate computations with totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.*, 29:731–751, 2007.
[20] P. Koev. `http://www.math.sjsu.edu/∼koev/`
[21] T. Laudadio, N. Mastronardi, P. Van Dooren. Computing Gaussian quadrature rules with high relative accuracy. *Num. Algor.*, 2022, `https://doi.org/10.1007/s11075-022-01297-9`.
[22] `http://www.netlib.org/lapack/`
[23] D. P. Laurie. Questions related to Gaussian quadrature formulas and two-term recursions. In: W. Gautschi, G.H. Golub, G. Opfer (Eds.), *Computation and Application of Orthogonal Polynomials.* International Series of Numerical Mathematics, 131:133–144, Birkhauser, Basel.
[24] D. P. Laurie. Computation of Gauss-type quadrature formulas. *J. Comput. Appl. Math.*, 127:201–217, 2001.
[25] Z. Liu and A. Narayan. On the computation of recurrence coefficients for univariate orthogonal polynomials. *J. Sci. Comput.*, 88, Article number: 53, 2021.
[26] E. Mainar, J. M. Peña, and B. Rubio. Accurate computations with collocation and Wronskian matrices of Jacobi polynomials. *J. Sci. Comput.*, 87, Article number: 77, 2021.
[27] A. Marco and J. J. Martínez. Accurate computations with totally positive Bernstein-Vandermonde matrices. *Electron. J. of Linear Algebra*, 26:357–380, 2013.
[28] A. Marco and J. J. Martínez. A total positivity property of the Marchenko-Pastur law. *Electron. J. of Linear Algebra*, 30:106–117, 2015.
[29] A. Marco, J. J. Martínez, and R. Viaña. Error analysis, perturbation theory and applications of the bidiagonal decomposition of rectangular totally positive h-Bernstein-Vandermonde matrices. *Linear Algebra Appl.*, 613:377–392, 2021.
[30] J. J. Martínez. Polinomios ortogonales, cuadratura gaussiana y problemas de valores propios. In: Luis Español and Juan L. Varona (Eds.), *Margarita Mathematica en memoria de José Javier (Chicho) Guadalupe Hernández.* Servicio de Publicaciones, Universidad de La Rioja, Logroño, Spain, pp. 595–606, 2001.
[31] J. C. Mason and D. C. Handscomb. *Chebyshev Polynomials.* Chapman and Hall/CRC, New York, 2002.
[32] N. Mastronardi, H. Taeter and P. Van Dooren. On computing eigenvectors of symmetric tridiagonal matrices. In: Bini, D., Di Benedetto, F., Tyrtyshnikov, E., Van Barel, M. (Eds.), *Structured Matrices in Numerical Linear Algebra.* Springer INdAM Series, vol. 30, pp. 181–195, 2019.
[33] `https://gerard-meurant.pagesperso-orange.fr/`
[34] G. Meurant and A. Sommariva. Fast variants of the Golub and Welsch algorithm for symmetric weight function in Matlab. *Num. Algor.*, 67:491–506, 2014.
[35] G. Meurant and P. Tichý. On computing quadrature-based bounds for the A-norm of the error in conjugate gradients. *Numer. Algor.*, 62:163–191, 2013.
[36] B. N. Parlett. For tridiagonals T replace T with LDL^t. *J. Comput. Appl. Math.*, 123:117–130, 2000.
[37] B. N. Parlett and O. A. Marques. An implementation of the dqds algorithm (positive case). *Linear Algebra Appl.*, 309:217–259, 2000.
[38] A. Pinkus. *Totally Positive Matrices.* Cambridge University Press, Cambridge, UK, 2010.
[39] A. H. Stroud. *Numerical Quadrature and Solution of Ordinary Differential Equations.* Applied Mathematical Sciences, vol. 10, Springer-Verlag, New York, 1974.
[40] G. Szegö. *Orthogonal Polynomials, 4th Ed.* American Mathematical Society, Providence, RI, 1975.
[41] A. Townsend. The Race to Compute High-order Gauss-Legendre Quadrature. *SIAM News*, 48(2):1–3, 2015.
[42] A. Townsend, T. Trogdon, and S. Olver. Fast computation of Gauss quadrature nodes and weights on the whole real line. *IMA J. Numer. Anal.*, 36(1):337–358, 2016.
[43] H. S. Wilf. *Mathematics for the Physical Sciences.* Wiley, New York, 1962. (Reprinted in 1978 by Dover Publications, New York).