On computing the zeros of a class of Sobolev orthogonal polynomials

N. Mastronardi[1†], M. Van Barel[2†], R. Vandebril[2†], P. Van Dooren[3†]

[1*]Istituto per le Applicazioni del Calcolo "M. Picone", Consiglio Nazionale delle Ricerche, via Amendola 122/D, Bari, Italy.
[2*]Department of Computer Science, KU Leuven, Celestijnenlaan 200A - Bus 2402 3001, Leuven, Belgium.
[3*]Department of Mathematical Engineering, Catholic University of Louvain, Avenue Georges Lemaitre 4, Louvain-la-Neuve, Belgium.

Contributing authors: nicola.mastronardi@cnr.it; marc.vanbarel@kuleuven.be; raf.vandebril@kuleuven.be; paul.vandooren@uclouvain.be;
[†]The authors contributed equally to this work.

Abstract

A fast and weakly stable method for computing the zeros of a particular class of hypergeometric polynomials is presented. The studied hypergeometric polynomials satisfy a higher order differential equation and generalize Laguerre polynomials. The theoretical study of the asymptotic distribution of the spectrum of these polynomials is an active research topic. In this article we do not contribute to the theory, but provide a practical method to contribute to further and better understanding of the asymptotic behavior. The polynomials under consideration fit into the class of Sobolev orthogonal polynomials, satisfying a four–term recurrence relation. This allows computing the roots via a generalized eigenvalue problem. After condition enhancing similarity transformations, the problem is transformed into the computation of the eigenvalues of a comrade matrix, which is a symmetric tridiagonal modified by a rank–one matrix. The eigenvalues are then retrieved by relying on an existing structured rank based fast algorithm. Numerical examples are reported studying the accuracy, stability and conforming the efficiency for various parameter settings of the proposed approach.

Keywords: Sobolev orthogonal polynomials, zeros of polynomials, generalized eigenvalue problem, comrade matrices

AMS Classification: 33C20 , 65F15 , 65F35

1 Introduction

Sobolev orthogonal polynomials have been intensively studied in the last decades as a generalization of classical orthogonal polynomials. Whereas the orthogonality of classical orthogonal polynomials is typically defined via a measure and polynomial evaluations in the discrete setting, Sobolev orthogonal polynomials also take the derivative or higher order derivatives into account when defining the inner product. Considering Sobolev orthogonal polynomials, the overview of Marcellán and Xu [1] provides a recent survey of both theoretical and application oriented developments and can serve as a starting point for further study (see also the work of Marcellán, Pérez, and Piñar [2]). Van Buggenhout provides an overview of the connection between Sobolev orthogonal polynomials and inverse eigenvalue problems, which links to Krylov as well [3]. With respect to applications, we can refer to the work of Liu, Yu, Wang and Li [4] and of Yi, Wang and Li [5] who study the use of particular Sobolev bases for solving differential equations. The hypergeometric polynomials considered in this article are a particular type of Sobolev orthogonal polynomials linked to a higher order differential equation; for more details we refer to the work of Zagorodnyuk [6] and the references therein. The polynomials also arise in the study of integral transforms, more particularly Riemann Liouville fractional integrals [7, Chapter XIII, equation (5)].

The particular hypergeometric polynomials we consider in this article are defined as

$$\mathfrak{L}_n(x) = {}_2F_2(-n, 1;\ \alpha+1, \kappa+1;\ x), \quad \alpha, \kappa > -1,$$

where

$${}_2F_2\left(\begin{bmatrix}\alpha\\ \beta\end{bmatrix}, \begin{bmatrix}\gamma\\ \delta\end{bmatrix};\ z\right) = \sum_{i=0}^{\infty} \frac{(\alpha)_i(\beta)_i}{(\gamma)_i(\delta)_i}\frac{z^i}{i!}$$

is the generalized hypergeometric function and $(\mu)_i$ is the Pochhammer symbol (shifted factorial) defined by $(\mu)_i = \mu(\mu+1)\cdots(\mu+i-1)$, for $i \in \mathbb{Z}_+$ and $(\mu)_0 = 1$.

We consider nonnegative integer values of the parameter κ, implying that the the polynomials $\mathfrak{L}_n(x)$ satisfy the following Sobolev orthogonality conditions ($\delta_{n,m}$ stands for the dirac delta):

$$\int_0^{\infty} \sum_{i=0}^{\kappa} \left(\frac{\kappa!}{i!}\binom{\kappa}{i}x^i\right)^2 \mathfrak{L}_n^{(i)}(x)\mathfrak{L}_m^{(i)}(x)e^{-x}x^{\alpha}dx = \frac{\kappa!^2\Gamma(\alpha+n+1)}{\binom{n+\alpha}{n}^2 n!}\delta_{m,n}.$$

The hypergeometric polynomials $\mathfrak{L}_n(x)$ are a generalization of the Laguerre polynomials [6] and satisfy the following 4–term recurrence relation

$$\begin{cases} \mathfrak{L}_{-2}(x) = 0, \\ \mathfrak{L}_{-1}(x) = 0, \\ \mathfrak{L}_0(x) = \eta, \\ x(e_i\mathfrak{L}_i(x) + f_i\mathfrak{L}_{i-1}(x)) = a_i\mathfrak{L}_{i+1}(x) + b_i\mathfrak{L}_i(x) + c_i\mathfrak{L}_{i-1}(x) + d_i\mathfrak{L}_{i-2}(x), \end{cases} \tag{1}$$

with $i \in \mathbb{N}$, $\eta \in \mathbb{R} \setminus \{0\}$, and

$$\begin{cases} a_i = -(i+\alpha+1)(i+\kappa+1), \\ b_i = i(2i+\alpha+\kappa+1) + (i+\alpha+1)(i+\kappa+1), \\ c_i = -i(3i+\alpha+\kappa), \\ d_i = (i-1)i, \\ e_i = i+1, \\ f_i = -i. \end{cases} \tag{2}$$

Remark 1 Given α and κ, we observe that, for a given $\eta \neq 0$, the sequence $\{\mathfrak{L}_i(x)\}_{i=0}^{\infty}$ is uniquely defined. The constant polynomial is initialized as η and does not influence the roots nor the algorithms proposed for computing the zeros of $\mathfrak{L}_n(x)$. The parameter η emerges as a scaling in the recurrence relation (2).

The location of the zeros of (general) hypergeometric polynomials $\mathfrak{L}_n(x)$, and in particular the study of the asymptotic distribution is of significant interest. We refer to Zagorodnuyk [6], the references below, and the references therein for more information on the topic. There are various approaches that can be used to study the distribution: Srivastava, Zhou, and Wang [8, 9] study directly the recursion coefficients; Zhou, Li, and Xu use the associated integral representation [10]; one can also exploit the link with other polynomials such as the Bessel functions, which is due to Bracciali and Moreno-Balcázar [11]. More generally applicable results are by Kuijlaars and Martínez-Finkelshtein who study the asymptotic distribution of the roots of Jacobi polynomials[12]; and Kuijlaars and Van Assche study the behavior of matrices in an asymptotic fashion [13]. The latter article [13] fits with the research in this article in the sense that we will provide an algorithm for efficiently computing the roots of matrices linked to hypergeometric polynomials. To the best of our knowledge the asymptotic eigenvalue distribution of this particular type of polynomials has not been found yet. Driver and Möller provide intervals in which the real zeros will lie, and they provide some numerical evidence linking the zeros to Cassini curves [14]. Boggs and Duren show that the zeros, for particular ranges of parameters, lie in a half plane and cluster on a particular loop of a lemniscate [15]. Duren and Guillou [16] refined this study using computer graphics.

In this article we describe a weakly stable algorithm for computing the zeros of $\mathfrak{L}_n(x)$. The algorithm has an $\mathcal{O}(n^2)$ computational complexity and uses $\mathcal{O}(n)$ memory. The eigenvalue problem is solved via a structure preserving and QR based method.

More information on QR algorithms can be found in Watkins' book [17], and structure preserving eigenvalue algorithms are discussed in the book of Mastronardi, Van Barel, and Vandebril [18]. To compute the roots of hypergeometric polynomials we first transform the generalized eigenvalue problem to a classical eigenvalue problem, which is then scaled to get a so called Comrade matrix. Comrade matrices, which are essentially symmetric tridiagonal plus spike matrices, are well suited for structure preserving QR iterations. Various algorithms that execute this task are readily available, see Eidelman, Gemignani, and Gohberg [19], Del Corso and Vandebril [20], and Casulli and Robol [21]. We will use this last one in this article, because it is theoretically proven to be the most reliable one.

The paper is organized as follows. In Section 2 it is shown that the zeros of $\mathfrak{L}_n(x)$ can be computed as the eigenvalues of a generalized eigenvalue problem associated with the pencil $xB_n - A_n$, where B_n and A_n are both banded. Furthermore we show how to transform the pencil to a classical eigenvalue problem of a comrade matrix, that is a symmetric tridiagonal matrix modified by a rank–one matrix. Section 3 presents the basic principles of the QR based, structure preserving algorithm for computing the eigenvalues. Numerical tests are reported in Section 4, where some rules of thumb for reliable use of the proposed method are derived. The conclusions are to be found in Section 5.

2 Two matrix eigenvalue problems

In this section we show that the zeros of $\mathfrak{L}_n(x)$ can be computed as the solution of two matrix eigenvalue problems: a generalized and a classical eigenvalue problem. If we write the 4-terms recurrence relation (1) for computing the polynomials $\mathfrak{L}_n(x)$ in matrix form, we obtain

$$(xB_n - A_n)\mathcal{L}_n(x) = a_{n-1}\mathfrak{L}_n(x)\mathbf{e}_n,$$

which we rewrite as

$$(xB_{n,n+1} - A_{n,n+1}) \begin{bmatrix} \mathcal{L}_n(x) \\ \mathfrak{L}_n(x) \end{bmatrix} = \mathbf{0},$$

where

$$A_n = \begin{bmatrix} b_0 & a_0 & & & \\ c_1 & b_1 & a_1 & & \\ d_2 & c_2 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-2} & a_{n-2} \\ & & d_{n-1} & c_{n-1} & b_{n-1} \end{bmatrix}, \quad B_n = \begin{bmatrix} e_0 & & & & \\ f_1 & e_1 & & & \\ & \ddots & \ddots & & \\ & & f_{n-2} & e_{n-2} & \\ & & & f_{n-1} & e_{n-1} \end{bmatrix},$$

and

$$\mathcal{L}_n(x) = \begin{bmatrix} \mathfrak{L}_0(x) \\ \mathfrak{L}_1(x) \\ \mathfrak{L}_2(x) \\ \vdots \\ \mathfrak{L}_{n-2}(x) \\ \mathfrak{L}_{n-1}(x) \end{bmatrix}, \; B_{n,n+1} = [B_n \mid \mathbf{0}], \text{ and } A_{n,n+1} = [A_n \mid a_{n-1}\mathbf{e}_n].$$

Therefore, $x^\star$ is a zero of $\mathfrak{L}_n(x)$ if and only if $x^\star$ is a generalized eigenvalue of the pencil

$$xB_n - A_n. \tag{3}$$

The following theorem states that the zeros of $\mathfrak{L}_n(x)$ are also the eigenvalues of a symmetric tridiagonal matrix plus a rank–one matrix. In practice it is often advised, for reasons of numerical reliability, to solve the generalized eigenvalue problem (3) directly. In this particular setting, however, we will see that the problem is heavily structured and allows us to transform it to a structured classical eigenvalue problem.

Theorem 1 *The matrix B_n is nonsingular and*

$$B_n^{-1} = D_n^{-1} M_n, \; D_n = \begin{bmatrix} 1 & & & & \\ & 2 & & & \\ & & \ddots & & \\ & & & n-1 & \\ & & & & n \end{bmatrix}, \; M_n = \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ \vdots & \dots & \ddots & & \\ 1 & 1 & \cdots & 1 & \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}. \tag{4}$$

Moreover,

$$X_n = B_n^{-1} A_n = \begin{bmatrix} \hat{v}_0 & \hat{a}_0 & & & \\ \hat{v}_1 & \hat{b}_1 & \hat{a}_1 & & \\ \hat{v}_2 & \hat{c}_2 & \ddots & \ddots & \\ \vdots & & \ddots & \hat{b}_{n-2} & \hat{a}_{n-2} \\ \hat{v}_{n-1} & & & \hat{c}_{n-1} & \hat{b}_{n-1} \end{bmatrix} \tag{5}$$

with

$$\begin{array}{lll} \hat{a}_{i-2} & = -\frac{(i-1+\alpha)(i-1+\kappa)}{i-1}, & i = 2,3,\dots,n \\ \hat{b}_{i-1} & = 2i - 1 + \alpha + \kappa, & i = 2,3,\dots,n, \\ \hat{c}_i & = -i, & i = 2,3,\dots,n-1, \\ \hat{v}_0 & = (\alpha+1)(\kappa+1), & \\ \hat{v}_1 & = \frac{\alpha\kappa}{2} - 1, & \\ \hat{v}_i & = \frac{\alpha\kappa}{i+1}, & i = 2,3,\dots,n-1. \end{array}$$

Proof Consider a nonsingular bidiagonal matrix, with all τ's and υ's different from zero:

$$B = \begin{bmatrix} \tau_1 & & & & \\ \upsilon_1 & \tau_2 & & & \\ & \ddots & \ddots & & \\ & & \upsilon_{n-2} & \tau_{n-1} & \\ & & & \upsilon_{n-1} & \tau_n \end{bmatrix}.$$

The inverse of this matrix is a so-called semiseparable matrix [22, Th. 4.5]. Because the υ's are nonzero, the inverse is the lower triangular part of a rank one matrix. We get that

$$B^{-1} = M^{-1}N,$$

where M and N are as follows

$$M = \begin{bmatrix} \tau_1 & & & & \\ & \tau_2 & & & \\ & & \ddots & & \\ & & & \tau_{n-1} & \\ & & & & \tau_n \end{bmatrix},$$

$$N = \begin{bmatrix} 1 & & & & & \\ (-\frac{\upsilon_1}{\tau_1}) & 1 & & & & \\ (-\frac{\upsilon_1}{\tau_1})(-\frac{\upsilon_2}{\tau_2}) & (-\frac{\upsilon_2}{\tau_2}) & 1 & & & \\ (-\frac{\upsilon_1}{\tau_1})(-\frac{\upsilon_2}{\tau_2})(-\frac{\upsilon_3}{\tau_3}) & (-\frac{\upsilon_2}{\tau_2})(-\frac{\upsilon_3}{\tau_3}) & (-\frac{\upsilon_3}{\tau_3}) & 1 & & \\ \vdots & & & \ddots & \ddots & \\ (-\frac{\upsilon_1}{\tau_1})\cdots(-\frac{\upsilon_{n-1}}{\tau_{n-1}}) & (-\frac{\upsilon_2}{\tau_2})\cdots(-\frac{\upsilon_{n-1}}{\tau_{n-1}}) & (-\frac{\upsilon_3}{\tau_3})\cdots(-\frac{\upsilon_{n-1}}{\tau_{n-1}}) & \cdots & (-\frac{\upsilon_{n-1}}{\tau_{n-1}}) & 1 \end{bmatrix}.$$

Hence, Equation (4) follows taking the correct diagonal and subdiagonal values of B (see (2)) into account.

For part two of the proof we consider first the matrix A_n. The ith column of A_n, $i = 2, \ldots, n$, is given by

$$\begin{bmatrix} \mathbf{0}_{i-2} \\ a_{i-2} \\ b_{i-1} \\ c_i \\ d_{i+1} \\ \mathbf{0}_{n-i-2} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{i-2} \\ -(i-1+\alpha)(i-1+\kappa) \\ (i-1)(2i-1+\alpha+\kappa)+(i+\alpha)(i+\kappa) \\ -i(3i+\alpha+\kappa) \\ i(i+1) \\ \mathbf{0}_{n-i-2} \end{bmatrix}.$$

Hence,

$$\begin{array}{lllll} \hat{a}_{i-2} & = B_n^{-1}(i-1,:)A_n(:,i) & = \frac{a_{i-2}}{i-1} & = -\frac{(i-1+\alpha)(i-1+\kappa)}{i-1}, & \\ \hat{b}_{i-1} & = B_n^{-1}(i,:)A_n(:,i) & = \frac{a_{i-2}+b_{i-1}}{i} & = 2i-1+\alpha+\kappa, & \\ \hat{c}_i & = B_n^{-1}(i+1,:)A_n(:,i) & = \frac{a_{i-2}+b_{i-1}+c_i}{i+1} & = -i, & \\ \hat{v}_0 & = B_n^{-1}(1,:)A_n(:,1) & = b_0 & = (\alpha+1)(\kappa+1), & \\ \hat{v}_1 & = B_n^{-1}(2,:)A_n(:,1) & \frac{b_0+c_1}{2} & = \frac{\alpha\kappa}{2} - 1, & \\ \hat{v}_i & = B_n^{-1}(i,:)A_n(:,1) & \frac{b_0+c_1+d_2}{i} & = \frac{\alpha\kappa}{i+1}, \quad i = 2, 3, \ldots, n-1. & \end{array} \tag{6}$$

Moreover,

$$X(i,j) = 0 \text{ if } i = 1, \ldots, n-2, \;\; i+2 < j \leq n$$

since D_n and M_n are lower triangular and A_n is lower Hessenberg.

Finally, for $j = 2, \ldots, n, \quad j+2 \leq i \leq 4$,

$$X(i,j) \; = \; B_n^{-1}(i,:)A_n(:,j) = \frac{1}{i}[\underbrace{1, \; \ldots, \; 1}_{i}, \underbrace{0, \; \ldots, \; 0}_{n-i}] \begin{bmatrix} \mathbf{0}_{j-2} \\ a_{j-2} \\ b_{j-1} \\ c_j \\ d_{j+1} \\ \mathbf{0}_{n-j-2} \end{bmatrix}$$

$$= \frac{a_{j-2} + b_{j-1} + c_j + d_{j+1}}{i} = 0$$

This concludes the proof by stating that A_n satisfies (5) $\square$

The matrix X_n is nothing else than a tridiagonal matrix plus a spike in the first column. Since $\hat{a}_i < 0, \; i = 0, \ldots, n-2$, and $\hat{c}_i < 0, \; i = 2, \ldots, n-1$, it is always possible to construct a diagonal matrix $D_n = \operatorname{diag}(\delta_1, \ldots, \delta_n)$ of order n such that $C_n = D_n X_n D_n^{-1}$ has its main tridiagonal part symmetric, which means that C_n without the spike is symmetric, or in other words the super- and subdiagonal are equal. This matrix is called a comrade matrix. The scaling allows us thus to write C_n as the sum of a symmetric tridiagonal plus rank one matrix $T_n + \mathbf{w}_1 \mathbf{e}_1^T$ with T_n symmetric and $\mathbf{w}_1 \mathbf{e}_1^T$ the spike:

$$C_n = D_n X_n D_n^{-1} = \left[\begin{array}{c|ccccc} \tilde{v}_0 & \tilde{a}_0 & & & & \\ \hline \tilde{a}_0 & \hat{b}_1 & \tilde{a}_1 & & & \\ \tilde{v}_2 & \tilde{a}_1 & \ddots & \ddots & & \\ \vdots & & \ddots & \hat{b}_{n-2} & \tilde{a}_{n-2} & \\ \tilde{v}_{n-1} & & & \tilde{a}_{n-2} & \hat{b}_{n-1} & \end{array}\right] = T_n + \mathbf{w}_1 \mathbf{e}_1^T, \tag{7}$$

with

$$T_n = \begin{bmatrix} \tilde{v}_0 & \tilde{a}_0 & & & \\ \tilde{a}_0 & \hat{b}_1 & \tilde{a}_1 & & \\ & \tilde{a}_1 & \ddots & \ddots & \\ & & \ddots & \hat{b}_{n-2} & \tilde{a}_{n-2} \\ & & & \tilde{a}_{n-2} & \hat{b}_{n-1} \end{bmatrix}, \quad \mathbf{w}_1 = \begin{bmatrix} 0 \\ \delta_2 \tilde{v}_1 - \tilde{a}_0 \\ \delta_3 \tilde{v}_2 \\ \vdots \\ \delta_n \tilde{v}_{n-1} \end{bmatrix}, \quad \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

The coefficients $\tilde{a}_i, \; i = 0, \ldots, n-2$, and $\delta_i, i = 1, \ldots, n$ can be computed recursively as follows. Initialize $\delta_1 = 1$ and $\delta_2 = 1$, then we can use the following iteration to compute the remaining values for $i = 3, \ldots, n$:

$$\delta_i = \delta_{i-1} \sqrt{\hat{a}_{i-2} / \hat{c}_{i-1}} \quad \text{and} \quad \tilde{a}_{i-2} = \sqrt{\hat{a}_{i-2} \hat{c}_{i-1}}. \tag{8}$$

Taking the values of $\hat{a}_i$ and $\hat{c}_i$ from (6), we can rewrite (8) as follows, for $i = 2, \ldots, n$:

$$\delta_{i+1} = \delta_i \sqrt{\frac{i+\alpha}{i} \frac{i+\kappa}{i}} \quad \text{and} \quad \tilde{a}_{i-1} = \sqrt{(i+\alpha)(i+\kappa)}.$$

We want to remark that care has to be taken when balancing a matrix for eigenvalue computations [23, 24]. In the numerical experiments we illustrate that balancing increases the sensitivity of the problem.

3 Computation of the zeros of $\mathfrak{L}_n(x)$

As shown, in Section 2, the zeros of $\mathfrak{L}_n(x)$ are the eigenvalues of a symmetric tridiagonal matrix T_n plus a rank–one modification $\mathbf{w}_1\mathbf{e}_1^T$. The eigenvalues of such matrices can be computed in a fast and backward stable fashion, i.e., with $\mathcal{O}(n^2)$ computational complexity, where n is the size of the matrix, by variants of the QR algorithm exploiting the structure of the involved matrices. In particular, to compute the eigenvalues of (7), we consider the algorithm by Casulli and Robol [21], a structure preserving variant of the QR algorithm.

The classical QR algorithm was originally developed by Francis and Kublanovskaya [25, 26], and is described in detail in Watkins [17]. Essentially, the QR algorithm computes the Schur decomposition $X = QUQ^T$, of a given input matrix $X \in \mathbb{R}^{n\times n}$, where U is upper triangular and Q orthogonal[1]. The eigenvalues are revealed on the main diagonal of the upper triangular matrix U. Of course, the algorithm is iterative in nature; it takes thus several iterations before the factorization is –numerically– achieved. Essentially, one starts with $X^{(0)} := X$, and at each iteration, called QR-step, applies an orthogonal similarity transformation to get $X^{(i)} = (Q^{(i-1)})^T X^{(i-1)} Q^{(i-1)}$, $i = 1, 2, \ldots$.

Every QR-step requiring $\mathcal{O}(n^3)$ floating point operations, bringing the matrix $X^{(i)}$ closer and closer to upper triangular form, until we are satisfied and have achieved a certain precision. To lower the computational cost, the matrix X is first transformed into a similar upper Hessenberg matrix, i.e., a matrix with all the entries equal to zero below the first subdiagonal.

The QR algorithm then starts iterating on this Hessenberg matrix, preserving the Hessenberg structure throughout all the iterations, and making the elements of the first subdiagonal closer and closer to zero. If applied to a Hessenberg matrix, each QR–step requires $\mathcal{O}(n^2)$ floating point operations, and the QR algorithm exibits $\mathcal{O}(n^3)$ computational complexity.

The QR algorithm can be similarly implemented considering lower Hessenberg matrices instead, i.e., matrices with all the entries equal to zero above the first superdiagonal. In this case, the sequence of matrices generated by the QR algorithm converges to a lower triangular matrix[2].

In case the involved matrix is (7), the variant of the QR algorithm developed by Casulli and Robol [21] exploits the structure of the matrix, requiring $\mathcal{O}(n)$ floating point operations for each QR-step, and $\mathcal{O}(n^2)$ overall computational complexity.

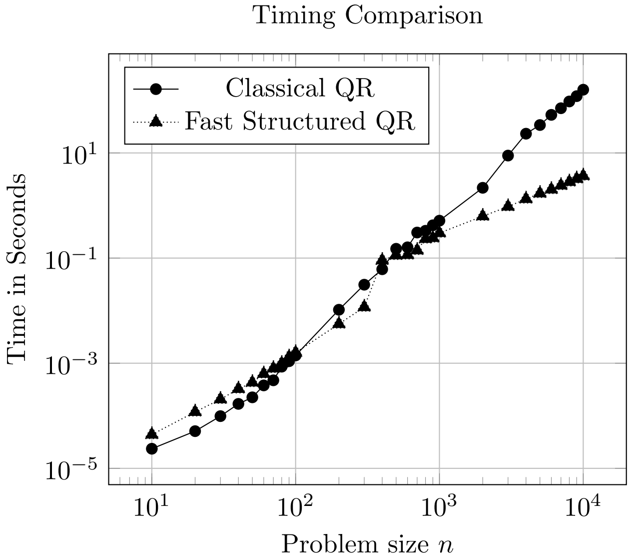
Let us shortly describe one QR-step of the latter algorithm. The involved matrix (7) has the form $C_n = T_n + \mathbf{w}_1\mathbf{e}_1^T$, i.e., a symmetric tridiagonal T_n plus the rank–one matrix $\mathbf{w}_1\mathbf{e}_1^T$. Therefore, C_n is already in lower Hessenberg form.

---

[1]For simplicity, we have restricted ourselves to real matrices and orthogonal similarity transformations. All of this can be generalized to the complex setting without loss of generality.

[2]This can be achieved easily by working on the transpose as the transposing a matrix does not change its eigenvalues.

Fig. 1 Comparison of timings in seconds between the classical order n^3 QR algorithm and the fast n^2 structured one.



Let $C_n^{(0)} = C_n$. Applying one QR-step to $C_n^{(0)}$, we get

$$C_n^{(1)} = {Q^{(0)}}^T C_n^{(0)} Q^{(0)} = {Q^{(0)}}^T T_n Q^{(0)} + ({Q^{(0)}}^T \mathbf{w}_1)\,({Q^{(0)}}^T \mathbf{e}_1)^T.$$

Clearly, ${Q^{(0)}}^T T_n Q^{(0)}$ is again symmetric, and $({Q^{(0)}}^T \mathbf{w}_1)\,({Q^{(0)}}^T \mathbf{e}_1)^T$ again of rank one form. Also the matrix $C_n^{(1)}$ will again, by construction, be in lower Hessenberg form. Moreover, the orthogonal matrix $Q^{(0)}$ is the product of $n-1$ Givens rotations. Hence, the computation of $C_n^{(1)}$ requires $\mathcal{O}(n)$ floating point operations.

Furthermore, these three properties, which are, symmetry, rank–one, and Hessenberg structure, will be maintained throughout all the iterations. This allows to develop a cheap, $\mathcal{O}(n)$ storage scheme for storing the matrices. As a consequence, a structure preserving QR algorithm will only take $\mathcal{O}(n^2)$ floating point operations, instead $\mathcal{O}(n^3)$, for computing all the eigenvalues of C_n.

Figure 1 shows the time in seconds for running the classical eigenvalue solver and the fast one. Starting from size 1000 the fast algorithm clearly wins. The experiments were done in Matlab, where the fast algorithm is encoded via Mex-files. The figure is merely to illustrate that the fast algorithm becomes significantly faster than the classical one at a certain point; we do not claim anything on the crossover point, this is highly depending on the programming language/implementations used.

4 Numerical experiments

In this section we will compare manners to compute the eigenvalues and compare them with respect to their numerical stability. We will use the fastest method of the three to examine some properties of the eigenvalues, i.e., the zeros of the hypergeometric polynomials

$$ {}_2F_2(-n, 1;\ \alpha + 1, \kappa + 1;\ x), \quad \alpha, \kappa \geq -1. $$

The three algorithms are the following ones:

- The first algorithm takes the unsymmetrized matrix X of size n for a given value of α and κ and solves the corresponding eigenvalue problem using the classical, not structure exploiting QR algorithm. For our purpose we have used the Matlab solver `eig`.
- The second algorithm uses the symmetrized Comrade matrix C as input and solves the corresponding eigenvalue problem using the non structured QR algorithm. Again we use the Matlab solver `eig`.
- The third algorithm uses the symmetrized matrix C as input and relies on a fast structure preserving QR algorithm. We opted to take the fast algorithm of Casulli and Robol [21].

We remark that we did not turn off the balancing option of the Matlab solver `eig`, which means that a particular balancing strategy is applied in order to enhance the accuracy. The fast algorithm does not allow for additional balancing. We will refer to the three approaches as Algorithms 1,2, and 3 respectively.

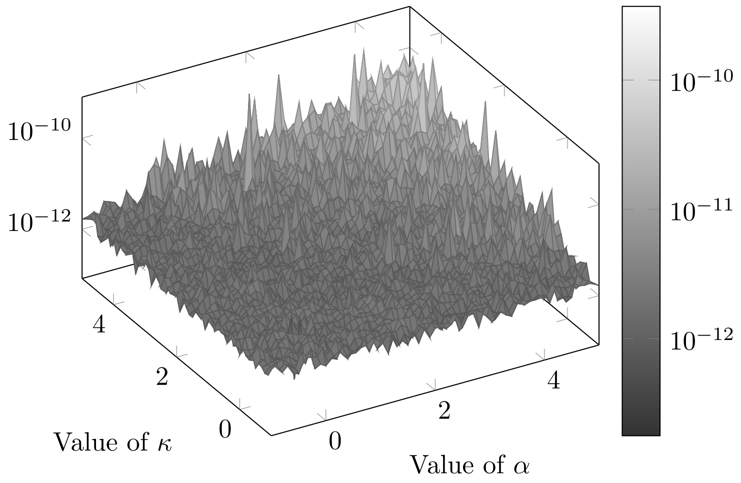
4.1 Stability analysis

To study the stability of these three algorithms we work in double precision and in multiple precision to construct the matrices and compute the eigenvalues. Figure 2 shows the maximum absolute error of the eigenvalues for size $n = 100$ and α and κ values taken from -1 to 5 by steps of 0.1. We assume the eigenvalues computed in multiple precision to be correct and determine the error based on those. In this first test we only considered the non-structured eigenvalue solver on the unsymmetrized problem, thus Algorithm 1. One can see that this error increases most at the diagonal of the α-κ plane. Hence we take $\alpha = \kappa$ in the next figures.

In Figure 3, the four subplots show this error for $\alpha = \kappa$ and for sizes $n = 100, 300, 400$ and 1000 for the three algorithms. Note that the legend is the same for all figures, but the y-axis scale, depicting the error, varies. We compare this absolute error with the product of the norm of the matrix X, the machine precision ϵ and the maximum of the condition number of the eigenvalues for X, and we do the same for C. The two latter measures are a prediction on the upper bound of the error to be expected. The condition number of an eigenvalue is $\frac{1}{w^T v}$ where v and w are the right and left eigenvectors of unit norm corresponding to the eigenvalue.

Algorithm 1 and 2 that use `eig` of Matlab use balancing by default. Their behavior is similar to the error measure for the unsymmetrized matrix because this is already a well-balanced matrix with respect to the eigenvalue problem. Because the fast algorithm is not using balancing, one could expect a behavior following the error measure for the symmetrized matrix but it behaves much better when $n < 400$. For higher

Fig. 2 Maximum absolute error on the eigenvalues for the non-structured QR algorithm on the nonsymmetrized matrix X.



values of n it starts to follow the error measure for the symmetrized matrix, which is very ill-balanced with respect to the eigenvalue problem. One can conclude that the three algorithms are weakly stable for their corresponding problem.

Summarizing we get the following result. For $\alpha = \kappa$ between -1 and 2.5 the three methods are comparable with respect to accuracy but if $n > 1000$ algorithm 3 becomes faster than the other two when n grows. For $\alpha = \kappa$ between 2.5 and 5 and $n < 400$ the fast algorithm is preferable because it gives the highest accuracy. However, if $n >= 400$ the other two algorithms are the best where `eig` applied to X is slightly better compared to C.

4.2 Range of stability for the fast algorithm

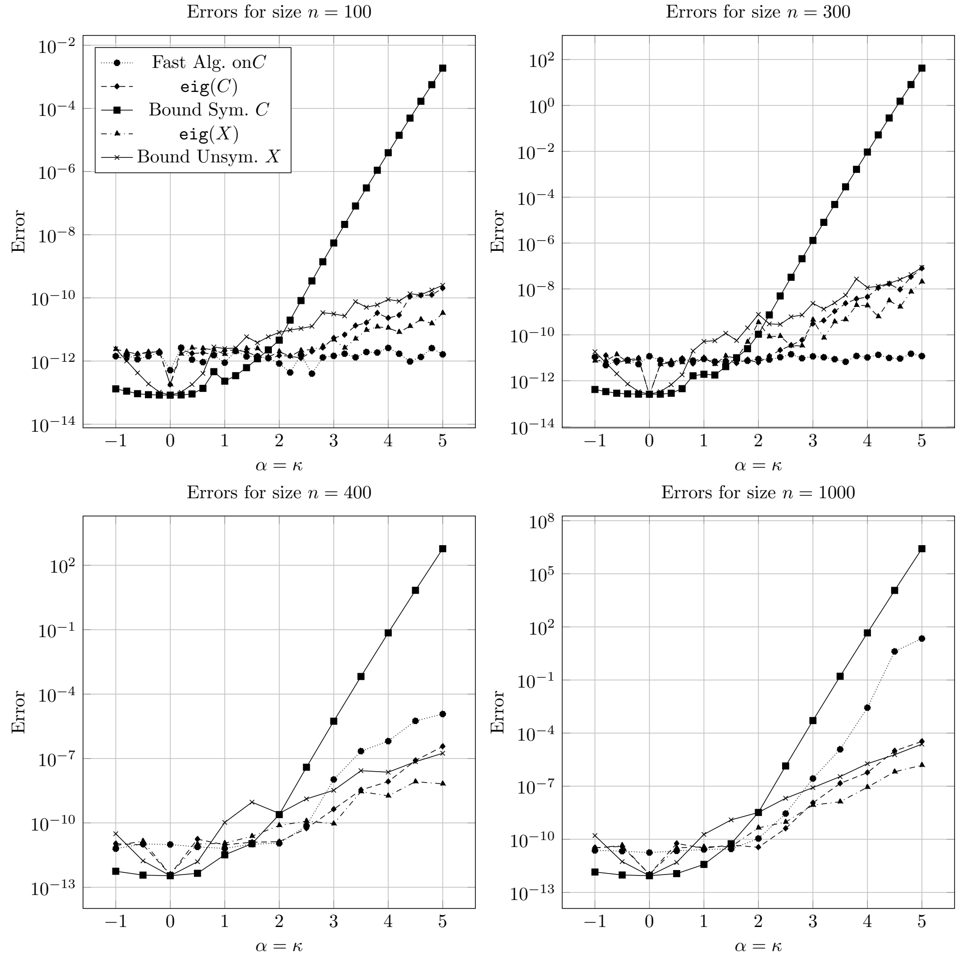
For the fast algorithm, one can expect visible changes in the plot of the spectrum for values of $\alpha = \kappa$ larger than 4, due to the difficulties of solving the problem For $\alpha = \kappa = 4.4$ and $n = 1000$ the eigenvalues computed in double and in multiple precision are plotted Figure 4. Note that for double precision, due to the ill-conditioning of the problem, 4 eigenvalues are moved away from the circular region of the eigenvalues.

We can conclude that the fast algorithm 3 can be used to study the behaviour of the eigenvalues depending on n, α and κ for values of α and κ smaller than 4 or 5.

4.3 Separation between complex and real spectrum

In Figure 5 the division lines between the regions in the $\alpha - \kappa$-plane are indicated where all the eigenvalues are real and where there are some of the eigenvalues that are non-real. We show this division line when the size of the matrix is 10, 100 or 1000. For α or κ equal to -1 or 0 all eigenvalues are real. In the bottom left region all eigenvalues are real while in the other 3 regions there are non-real eigenvalues. We remark that the curve for $n = 10000$ could only be computed by using the fast algorithm.

Fig. 3 Comparison of the absolute error of the three algorithms and two additional curves representing upper bounds for both nonsymmetric and symmetric solvers. Note that the legend holds for all subfigures, but the y-axis scale differs significantly.



4.4 Growth of the spectrum

Figure 6 shows the maximum of the real part of the eigenvalues which is also the maximum real eigenvalue for sizes $n = 10 : 10 : 100;\ 200 : 100 : 1000;\ 2000 : 1000 : 10000$. The different lines correspond to different values for α and κ. The parameter α is taken from $-1 : 1 : 5$ as well as the parameter κ. For small values of the size of the matrix n there is some variation depending on the values of α and κ while this becomes smaller for increasing values of n.

Fig 7 we show the maximum of the absolute value of the imaginary part of the eigenvalues for the size of the matrix $n = 1000$ and for α and κ values taken from $-1 : 0.1 : +4$.

Fig. 4 High precision and double precision plot of part of the spectrum, to illustrate the ill-conditioning. The eigenvalues computed in double precision deviate from the correct ones.

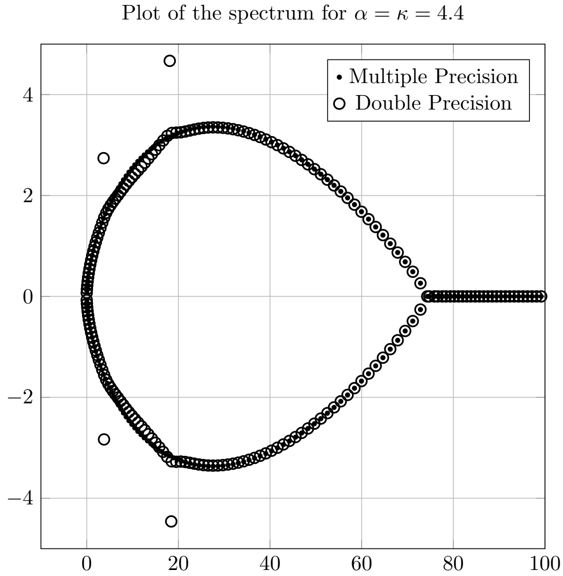


Figure 8 shows the minimum of the real part of all the eigenvalues where the size of the matrix n is taken from $10 : 10 : 100$; $200 : 100 : 1000$; $2000 : 1000 : 10000$ while α and κ are taken from $0 : 1 : 3$ where we have excluded the value α or κ equal to -1. In this case the minimum of the real part of the eigenvalues is 0 corresponding to the eigenvalue 0.

5 Conclusions

In this article a novel fast method was proposed to analyze and compute the spectrum of a particular class of hypergeometric polynomials. Numerical experiments illustrated the viability of the approach for particular ranges of α and κ and some numerical analysis of the spectrum of the polynomials was presented.

Acknowledgements

Nicola Mastronardi is member of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM). The work of Nicola Mastronardi was partly supported by MIUR, PROGETTO DI RICERCA DI RILEVANTE INTERESSE NAZIONALE (PRIN) 20227PCCKZ "Low–rank Structures and Numerical Methods in Matrix and Tensor Computations and their Application", Università degli Studi di BOLOGNA CUP J53D23003620006. The work of Marc Van Barel and Raf Vandebril was partially supported by the Research Council KU Leuven (Belgium), project C16/21/002 (Manifactor: Factor Analysis for Maps into Manifolds) and by the Fund for Scientific Research – Flanders (Belgium), projects G0A9923N (Low rank tensor approximation techniques for up- and downdating of massive online time

Fig. 5 Separation lines in the α κ-plane to distinguish two regions. Below left the eigenvalues are real, crossing the line results in complex eigenvalues.

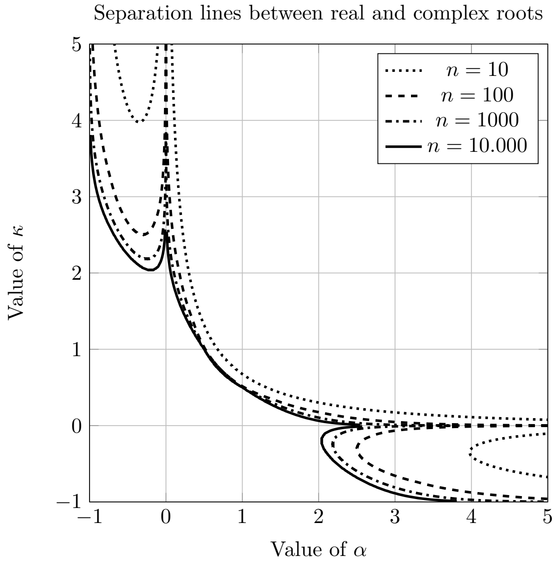
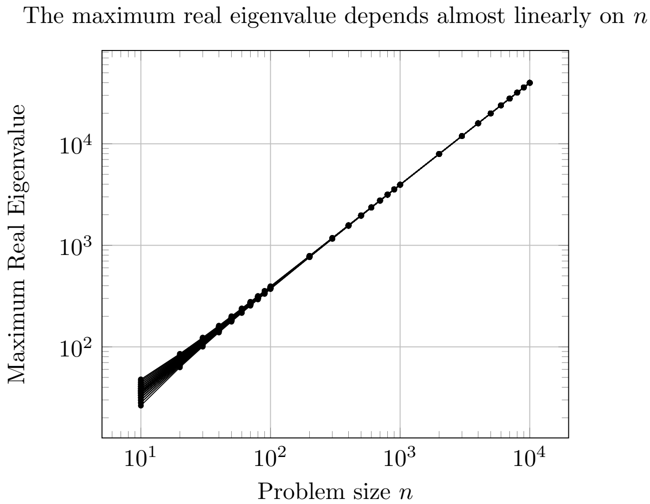


Fig. 6 Maximum real part of the eigenvalues.



series clustering) and G0B0123N (Short recurrence relations for rational Krylov and orthogonal rational functions inspired by modified moments).

The authors also wish to thank Prof. Zagorodnyuk, for swiftly responding and pointing to some references of interest for this article.

Fig. 7 Maximum absolute value of imaginary part of the eigenvalues.

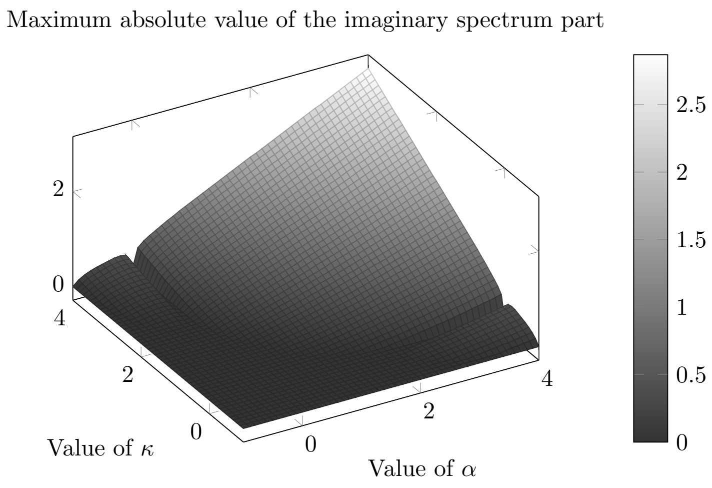
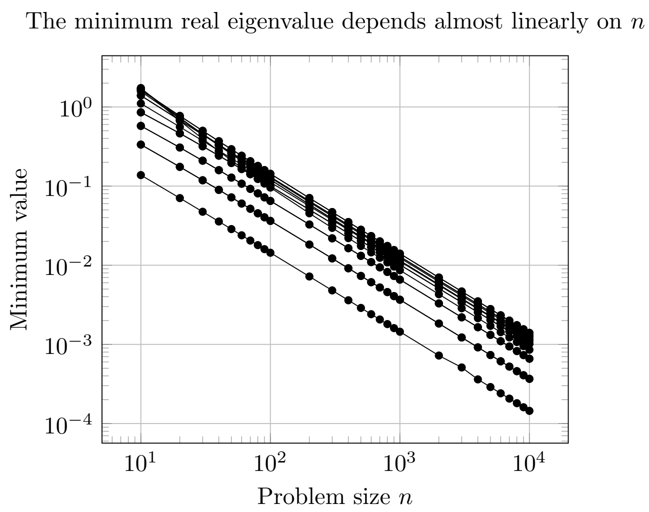


Fig. 8 The minimum real part of the spectrum.



References

[1] Marcellán, F. & Xu, Y. On Sobolev orthogonal polynomials. *Expo. Math.* **33**, 308–352 (2015).

[2] Marcellán, F., Pérez, T. E. & Piñar, M. A. Laguerre-Sobolev orthogonal polynomials. *J. Comput. Appl. Math.* **71**, 245–265 (1996).

[3] Van Buggenhout, N. On generating Sobolev orthogonal polynomials. *Numer. Math.* **155** (2023).

[4] Liu, Y., Yu, X.-H., Wang, Z.-Q. & Li, H.-Y. Hermite-sobolev orthogonal functions and spectral methods for second and fourth-order problems on unbounded domains. *Int. J. Comput. Math.* **96**, 950–970 (2019).

[5] Yu, X., Wang, Z. & Li, H. Jacobi Sobolev orthogonal polynoimals and spectral methods for elliptic boundary value problems. *Comm. Appl. Matho. Comput.* **1**, 283–308 (2019).

[6] Zagorodnyuk, S. M. On some classical type Sobolev orthogonal polynomials. *J. Approx. Theory* **250**, 105337 (2020).

[7] Erdélyi, A., Magnus, W., Oberhettinger, F. & Tricomi, F. G. *Tables of Integral Trasnforms* Vol. II (McGraw-Hill Book Company, Inc., New York, Toronto, London, 1954).

[8] Zhou, J.-R., Srivastava, H. M. & Wang, Z. G. Asymptotic distributions of the zeros of a family of hypergeometric polynomials. *Proc. Amer. Math. Soc.* **140**, 2333–2346 (2012).

[9] Srivastava, H. M., Zhou, J.-R. & Wang, Z.-G. Asymptotic distributions of the zeros of certain classes of hypergeometric functions and polynomials. *Math. Comp.* **80**, 1769–1784 (2011).

[10] Zhou, J.-R., Li, H. & Xu, Y. Asymptotic distribution of the zeros of a certain family of generalized hypergeometric polynomials. *Applicable Analysis* **103**, 2455–2464 (2023).

[11] Bracciali, C. F. & Moreno-Balcázar, J. J. On the zeros of a class of generalized hypergeometric polynomials. *Appl. Math. Comput.* **253**, 151–158 (2015).

[12] Kuijlaars, A. B. J. & Martínez-Finkelshtein, A. Strong asymptotics for Jacobi polynomials with varying nonstandard parameters. *J. Analyse Math.* **94**, 195–234 (2004).

[13] Kuijlaars, A. B. J. & Van Assche, W. The asymptotic zero distribution of orthogonal polynomials with varyng recurrence coefficients. *J. Approx. Theory* **99**, 167–197 (1999).

[14] Driver, K. & Möller, M. Zeros of the hypergeometric polynomials $f(-n, b;\ -2n;\ z)$. *J. Approx. Theory* **110**, 74–87 (2001).

[15] Boggs, K. & Duren, P. Zeros of hypergeometric functions. *Computational Methods and Function Theory* **1**, 275–287 (2001).

[16] Duren, P. L. & Guillou, B. J. Asymptotic properties of zeros of hypergeometric polynomials. *J. Approx. Theory* **111**, 329–343 (2001).

[17] Watkins, D. S. *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods* (SIAM, Philadelphia, USA, 2007).

[18] Vandebril, R., Van Barel, M. & Mastronardi, N. *Matrix Computations and Semiseparable Matrices, Volume II: Eigenvalue and Singular Value Methods* (Johns Hopkins University Press, Baltimore, Maryland, USA, 2008).

[19] Eidelman, Y., Gemignani, L. & Gohberg, I. C. Efficient eigenvalue computation for quasiseparable Hermitian matrices under low rank perturbation. *Numer. Algorithms* **47**, 253–273 (2008).

[20] Vandebril, R. & Del Corso, G. M. An implicit multishift QR-algorithm for

Hermitian plus low rank matrices. *SIAM J. Sci. Comput.* **32**, 2190–2212 (2010).

[21] Casulli, A. & Robol, L. Rank-structured QR for Chebyshev rootfinding. *SIAM J. Matrix Anal. Appl.* **42**, 1148–1171 (2021).

[22] Vandebril, R., Van Barel, M. & Mastronardi, N. *Matrix Computations and Semiseparable Matrices, Volume I: Linear Systems* (Johns Hopkins University Press, Baltimore, Maryland, USA, 2008).

[23] Lemonnier, D. & Van Dooren, P. Balancing regular matrix pencils. *SIAM J. Matrix Anal. Appl.* **28**, 253–263 (2006).

[24] Watkins, D. S. A case where balancing is harmful. *Electron. Trans. Numer. Anal.* **23**, 1–4 (2006).

[25] Francis, J. G. F. The QR Transformation a unitary analogue to the LR transformation–Part 1. *Comput. J.* **4**, 265–271 (1961).

[26] Kublanovskaya, V. N. On some algorithms for the solution of the complete eigenvalue problem. *USSR Computational Mathematics and Mathematical Physics* **1**, 637–657 (1962).

6 Uncited References

The following references were in the paper, but were not cited.

- O. Aberth, Iteration methods for finding all zeros of a polynomial simultaneously, Math. Comput. 27 (1973) 339–344.
- R. Adams, On hyperbolic polynomials with four–term recurrence and linear coefficients. Calcolo 57, Article number: 22 (2020). https://doi.org/10.1007/s10092-020-00373-7
- Q. Ai, H. Li, Z. Wang, Diagonalized Legendre spectral methods using Sobolev orthogonal polynomals for elliptic boundary value problems, Appl. Numer. Math. 127 (2018), 196-210.
- P. Althammer, Eine Erweiterung des Orthogonalitätsbegriffes bey Polynomen und deren Anwendung auf die beste Approximation, J. Reine Angew. Math. 211 (1962)192–204. In German.
- T. S. Chihara, An Introduction to Orthogonal Polynomials, Gordon and Breach, New York, 1978.
- R. Vandebril, G.M. Del Corso, A unification of unitary similarity transforms to compressed representations. Numer. Math. 119 (2011) 641–665.
- L. Fernández, F. Marcellán, T. E. Pérez, M. Piñar, Sobolev orthogonal polynomials and spectral methods in boundary value problems. Appl. Numer. Math. 200 (2024) 254-272.
- L. Fernández, F. Marcellán, T. E. Pérez, M. Piñar, Sobolev orthogonal polynomials for solving the Schrödinger equation with potentials $V(x) = x^{2k}$, $k \geqslant 1$. Coimbra Mathematical Texts. Springer. Cham. 2024. In press.
- G. H. Golub, J. H. Welsch, Calculation of Gauss quadrature rules, Math. Comp. 23 (1969), 221–230.
- G.H. Golub and C.F. Van Loan, *Matrix Computations,* 4th ed., Johns Hopkins University Press, Baltimore, 2013.
- L.W. Ehrlich, A modified Newton method for polynomials, *Comm. ACM* Vol.10:107–108, 1967.
- N.J. Higham, The Accuracy of Solutions to Triangular Systems, *SIAM Journal on Numerical Analysis*, Vol. 26(5) 1989 pp. 1252–1265.
- A. Iserles, P. E. Koch, S. P. Nørsett, J. M. Sanz–Serna, Orthogonality and approximation in a Sobolev space. In Algorithms for approximation, II (Shrivenham, 1988), 117-124, Chapman and Hall, London, 1990.
- A. Iserles, P. E. Koch, S. P. Nørsett, J. M. Sanz-Serna, On polynomials orthogonal with respect to certain Sobolev inner products. J. Approx. Theory 65 (2) (1991) 151–175.
- T. Laudadio, N. Mastronardi, P. Van Dooren, Computational aspects of simultaneous Gaussian quadrature, Numer. Algorithms
- T. Laudadio, N. Mastronardi, W. V. Assche, P. Van Dooren, A Matlab package computing simultaneous Gaussian quadrature rules for multiple orthogonal polynomials, J. Comput. Appl. Math. accepted.

- T. Laudadio, N. Mastronardi, P. Van Dooren, Fast and reliable algorithms for computing the zeros of Althammer polynomials, Appl. Numer. Math. 207 (2025) 210-221.
- H. G. Meijer, Determination of all coherent pairs, J. Approx. Theory 89 (3) (1997) 321–343.
- K. Tran, A. Zumba, Zeros of polynomials with four-term recurrence and linear coefficients, The Ramanujan Journal 55 (2021) 447–470. https://doi.org/10.1007/s11139-020-00263-0
- K. Tran, A. Zumba, Zeros of polynomials with four-term recurrence, Involve, 11(3), 501–518 (2018). https://doi.org/10.2140/involve.2018.11.501.